Long-Wen Chang
Wen-Nung Lie (Eds.)

LNCS 4319

Advances in Image and Video Technology

First Pacific Rim Symposium, PSIVT 2006
Hsinchu, Taiwan, December 2006
Proceedings

Springer

Lecture Notes in Computer Science 4319

Long-Wen Chang Wen-Nung Lie (Eds.)

Advances in Image and Video Technology

First Pacific Rim Symposium, PSIVT 2006
Hsinchu, Taiwan, December 10-13, 2006
Proceedings

Springer

Volume Editors

Long-Wen Chang
National Tsing Hua University
Department of Computer Science
Hsinchu, 300 , Taiwan
E-mail: lchang@cs.nthu.edu.tw

Wen-Nung Lie
National Chung Cheng University
Department of Electrical Engineering
Chia-Yi, 621 Taiwan
E-mail: ieewnl@ccu.edu.tw

Library of Congress Control Number: 2006937888

CR Subject Classification (1998): H.5.1, H.5, I.4, I.3, H.3-4, E.4

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-540-68297-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-68297-4 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11949534 06/3142 5 4 3 2 1 0

Preface

The conference of the 1st IEEE Pacific-Rim Symposium on Image and Video Technology (PSIVT 2006) was held at Hsinchu, Taiwan, Republic of China, on December 11–13, 2006. This volume contains papers selected for presentation at this conference. The aim of this conference was to bring together theoretical advances and practical implementations contributing to, or being involved in, image and video technology.

PSIVT 2006 featured a comprehensive program including tutorials, keynote and invited talks, oral paper presentations, and posters. We received 450 submissions from 22 countries and accepted 141 papers among those (i.e., defining an acceptance rate of 31.3%). The intention was to establish PSIVT as a top-quality series of symposia. Decisions were difficult sometimes, but we hope that the final result is acceptable to all involved.

Besides keynotes and invited talks, PSIVT 2006 offered 76 oral presentations and 58 posters, according to the proper registration of these papers by the defined deadline. We deeply appreciate the help of the reviewers, who generously spent their time to ensure a high-quality reviewing process. Useful comments were provided by reviewers, often quite detailed, and they certainly offered authors opportunities to improve their work not only for this conference, but also for future research.

We thank Springer's LNCS department and IEEE's Circuits and Systems Society for efficient contacts during the preparation of the conference and these proceedings. Their support is greatly appreciated. This conference would never have been successfully completed without the efforts of many people. We greatly appreciate the effort and the cooperation provided by our strong Organizing Committee. We would also like to thank all the sponsors for their considerable support including the National Tsing Hua University (NTHU), National Chung Cheng University (NCCU), The University of Auckland (UoA), National Science Council (NSC), Ministry of Education (MoE), Sunplus Technology Co., National Center for High-Performance Computing (NCHC), and Institute for Information Industry (III).

October 2006

Long-Wen Chang
Wen-Nung Lie

PSIVT 2006 Organization

Organizing Committee

General Co-chairs	Yung-Chang Chen (National Tsing Hua Univ., Taiwan) Reinhard Klette (The Univ. of Auckland, New Zealand)
Program Co-chairs	Long-Wen Chang (National Tsing Hua Univ., Taiwan) Wen-Nung Lie (National Chung Cheng Univ., Taiwan)
Local Arrangements Co-chairs	Chaur-Chin Chen (National Tsing Hua Univ., Taiwan) Shang-Hong Lai (National Tsing Hua Univ., Taiwan)
Publicity Chair	Chiou-Ting Hsu (National Tsing Hua Univ., Taiwan)
Finance Chair	Tai-Lang Jong (National Tsing Hua Univ., Taiwan)
Publication Chair	Rachel Chiang (National Chung Cheng Univ., Taiwan)
Tutorial and Invited Speaker Chairs	Chia-Wen Lin (National Chung Cheng Univ., Taiwan)
Award Co-chairs	Jin-Jang Leou (National Chung Cheng Univ., Taiwan) Chia-Wen Lin (National Chung Cheng Univ., Taiwan)
Exhibition Chair	Chung-Lin Huang (National Tsing Hua Univ., Taiwan)
Exhibition Co-chairs	Chung-Lin Huang (National Tsing Hua Univ., Taiwan) Charng-Long Lee (Sunplus Tech. Co., Taiwan)
Webmaster	Chao-Kuei Hsieh (National Tsing Hua Univ., Taiwan)
Steering Committee	Kap Luk Chan (Nanyang Technological Univ., Singapore) Yung-Chang Chen (National Tsing Hua Univ., Taiwan)

Yo-Sung Ho (Gwangju Institute of Science and Tech., Korea)
Reinhard Klette (The Univ. of Auckland, New Zealand)
Mohan M. Trivedi (Univ.of California at San Diego, USA)

Theme Co-chairs

3D Scene Modeling
Domingo Mery (Pontificia Universidad Catolica de Chile, Chile)
Long Quan (The Hong Kong Univ. of Science and Technology, Hong Kong)

Image Analysis
Chia-Yen Chen (National Chung Cheng Univ., Taiwan)
Kazuhiko Kawamoto (Kyushu Institute of Technology, Japan)

Intelligent Vision Applications
Kap Luk Chan (Nanyang Technological Univ., Singapore)
Chin-Teng Lin (National Chiao-Tung Univ., Taiwan)

Multimedia Compression and Transmission
Ramakrishna Kakarala (Avago Technologies, USA)
Shipeng Li (Microsoft Research Asia, China)

Multimedia Signal Processing
Yo-Sung Ho (Gwangju Institute of Science and Tech., Korea)
Ya-Ping Wong (Multimedia University, Malaysia)

Panoramic Imaging and Distributed Video Systems
Kenichi Kanatani (Okayama University, Japan)
Mohan M. Trivedi (Univ. of California at San Diego, USA)

Sensors Technologies
Charng-Long Lee (Sunplus Tech. Co., Taiwan)
Y. Tim Tsai (ITRI, Taiwan)

Visualization
Masa Takatsuka (The Univ. of Sydney, Australia)
Yangsheng Wang (CBSR, CASIA, China)

Program Committee

3D Scene Modeling

Chi-Fa Chen (I-Shou Univ., Taiwan)
Boris Escalante (Univ. Autonoma de Mexico, Mexico)
Andre Gagalowicz (INRIA, France)
Nancy Hitschfeld (Universidad de Chile, Chile)
Yi-Ping Hung (National Taiwan Univ., Taiwan)
Atsushi Imiya (Chiba Univ., Japan)
Nahum Kiryati (Tel Aviv Univ., Israel)
Brendan McCane (Univ. of Otago, New Zealand)
Domingo Mery (Pontificia Universidad Catolica de Chile, Chile)
Luis Pizarro (Saarland Univ., Germany)
Long Quan (The Hong Kong Univ. of Science and Technology, Hong Kong)
Fernando Rannou (Univ. Santiago de Chile, Chile)
Bodo Rosenhahn (MPI, Germany)
Luis Rueda (Univ. Concepcion, Chile)
Hideo Saito (Keio Univ., Japan)
Robert Valkenburg (Industrial Research Ltd., New Zealand)

Image Analysis

Ruey-Feng Chang (National Taiwan Univ., Taiwan)
Chia-Yen Chen (The Univ. of Auckland, New Zealand)
Sei-Wang Chen (National Taiwan Normal Univ., Taiwan)
Li Chen (Univ. of the District of Columbia, USA)
Hidekata Hontani (Nagoya Institute of Technology, Japan)
Norikazu Ikoma (Kyushu Institute of Technology, Japan)
Xiaoyi Jiang (Univ. of Münster, Germany)
Kazuhiko Kawamoto (Kyushu Institute of Technology, Japan)
Yukiko Kenmochi (French Natl. Center for Scientific Research, France)
Yoon Ho Kim (Mokwon Univ., Korea)
Anthony Maeder (CSIRO, ICT Centre, Australia)
Akira Nakamura (Hiroshima, Japan)
Hajime Nobuhara (Univ. of Tsukuba, Japan)
Nicolai Petkov (Groningen Univ., Netherlands)
Gerd Stanke (GFaI, Germany)
Akihiro Sugimoto (National Instutite of Informatics, Japan)
Yung-Nien Sun (National Cheng Kung Univ., Taiwan)
Toru Tamaki (Hiroshima Univ., Japan)
Evelyn L. Tan (Univ. of the Philippines, Quezon City, Philippines)
Petra Wiederhold (CINVESTAV, Mexico)
Huabei Zhou (Wuhan Univ., China)

Intelligent Vision Applications

Jacky Baltes (Univ. of Manitoba, Canada)
John Barron (Univ. of Western Ontario, Canada)
Chris Bowman (Industrial Research Ltd., New Zealand)
Thomas Braunl (The Univ. of Western Australia, Australia)
Kap Luk Chan (Nanyang Technological Univ., Singapore)
How-lung Eng (Institute of Infocomm. Research, Singapore)
Uwe Franke (DaimlerChrysler AG, Machine Perception, Germany)
Jessie Jin (Univ. of Newcastle, Australia)
Ron Kimmel (Technion, Israel)
Chin-Teng Lin (National Chiao Tung Univ., Taiwan)
Brian Lovell (The Univ. of Queensland, Australia)
Herbert Suesse (Jena Univ., Germany)
Shuicheng Yan (Univ. of Illinois at Urbana-Champaign, USA)
Su Yang (Fudan Univ., China)
Wei Yun Yau (Institute of Infocomm. Research, Singapore)

Multimedia Compression and Transmission

Mei-Juan Chen (National Dong-Hwa Univ., Taiwan)
Markus Flierl (Stanford Univ., USA)
Wenjen Ho (Institutes of Information Industry, Taiwan)
Ramakrishna Kakarala (Avago Technologies, USA)
Andreas Koschan (Univ. of Tennessee, Knoxville, USA)
Chang-Ming Lee (National Chung Cheng Univ., Taiwan)
Shipeng Li (Microsoft Research Asia, China)
Vincenzo Liguore (Ocean Logic Pty Ltd., Australia)
Yan Lu (Microsoft Research Asia, China)
Lei Ming (TVIA Inc., USA)
Philip Ogunbona (Univ. of Wollongong, Australia)
Volker Rodehorst (Tenchnische Universität Berlin, Germany)
Gary Sullivan (Microsoft Corporation, USA)
Alexis M. Tourapis (Dolby Corporation, USA)
Feng Wu (Microsoft Research Asia, China)
Chia-Hung Yeh (National Dong-Hwa Univ., Taiwan)
Lu Yu (Zhejiang Univ., China)
Bing Zeng (The Hong Kong Univ. of Science and Technology, Hong Kong)

Multimedia Signal Processing

Oscar Au (The Hong Kong Univ. of Science and Technology, Hong Kong)
Berlin Chen (National Taiwan Normal Univ., Taiwan)
Yo-Sung Ho (Gwangju Institute of Science and Tech., Korea)
JunWei Hsieh (Yuan-Ze Univ., Taiwan)
Hideaki Kimata (NTT, Japan)
Yung-Lyul Lee (Sejong Univ., Korea)

Xuelong Li (Univ. of London, UK)
Aljoscha Smolic (Fraunhofer-HHI, Germany)
Kwanghoon Sohn (Yonsei Univ., Korea)
Yeping Su (Thomson Co., USA)
Masayuki Tanimoto (Nagoya Univ., Japan)
Ya-Ping Wong (Multimedia Univ., Malaysia)
Marcel Worring (Univ. of Amsterdam, Netherlands)

Panoramic Imaging and Distributed Video Systems

Narendra Ahuja (Beckman Institute, Univ. of Illinois, USA)
Elli Angelopoulou (Stevens Inst. of Technology, USA)
Naoki Chiba (SANYO North America Corporation, USA)
Tomio Echigo (Osaka Univ., Japan)
Tarak Gandhi (CVRR, USA)
Fay Huang (National Yi-Lan Univ., Taiwan)
Kohsia Huang (CVRR, USA)
Naoyuki Ichimura (AIST, Japan)
Kenichi Kanatani (Okayama Univ., Japan)
Sangho Park (CVRR, USA)
Shmuel Peleg (Hebrew Univ., Israel)
Richard Radke (Rensselaer Polytechnic Institute, USA)
Nobutaka Shimada (Ritsumeikan Univ., Japan)
Mohan Trivedi (CVRR, USA)

Sensor Technologies

Anko Boerner (German Aerospace Center (DLR), Germany)
Oscal T.-C. Chen (National Chung Cheng Univ., Taiwan)
Chiou-Shann Fuh (National Taiwan Univ., Taiwan)
Herbert Jahn (German Aerospace Center (DLR), Germany)
Charng-Long Lee (Sunplus Inc., Taiwan)
C.F. Lin (Yuan Zhe Univ., Taiwan)
Y.T. Liu (Altek Inc., Taiwan)
Bruce MacDonald (The Univ. of Auckland, New Zealand)
John Morris (The Univ.of Auckland, New Zealand)
Ralf Reulke (Humboldt Univ., Germany)
Martin Scheele (German Aerospace Center (DLR), Germany)
Ewe Hong Tat (Multimedia Univ., Malaysia)
Y. Tim Tsai (ITRI, Taiwan)
Sheng-Jyh Wang (National Chiao Tung Univ., Taiwan)
David Yuen (Univ. of Southampton, UK)

Visualization

Thomas Buelow (Philips Research Lab., Germany)
Patrice Delmas (The Univ. of Auckland, New Zealand)
David Dagan Feng (Hong Kong Polytech. Univ., Hong Kong)

Richard Green (Univ. of Canterburyh, New Zealand)
Reinhard Koch (Christian-Albrechts-Univ. of Kiel, Germany)
Damon Shing-Min Liu (National Chung Cheng Univ., Taiwan)
Ngoc-Minh Le (HCMC Univ. of Technology, Vietnam)
Andrew Vande Moere (Univ. of Sydney, Australia)
Shigeru Owada (Sony CSL, Japan)
Masa Takatsuka (The Univ. of Sydney, Australia)
Matthias Teschner (Freiburg Univ., Germany)
Yangsheng Wang (CBSR, CASIA, China)
Michael Wilkinson (Groningen Univ., Netherlands)
Jason Wood (The Univ. of Leeds, UK)

Reviewers

Narendra Ahuja
Kiyoharu Aizawa
Elli Angelopoulou
Mohsen Ashourian
Oscar Au
Jacky Baltes
John Barron
Anko Boerner
Chris Bowman
Thomas Buelow
Hyeran Byun
Kap Luk Chan
Chuan-Yu Chang
I-Cheng Chang
Kevin Y.-J. Chang
Long-Wen Chang
Berlin Chen
Chaur-Chin Chen
Chi-Fa Chen
Chia-Yen Chen
Chih-Ming Chen
Ling-Hwei Chen
Jiann-Jone Chen
Li Chen
Mei-Juan Chen
Oscal T.-C. Chen
Sei-Wang Chen
Shu-Yuan Chen
Yong-Sheng Chen
Fang-Hsuan Cheng
Shyi-Chyi Cheng
Cheng-Chin Chiang
Rachel Chiang
Naoki Chiba
Yoon Sik Choe
Chun-Hsien Chou
Cheng-Hung Chuang
Jen-Hui Chuang
Pau-Choo Chung
Patrice Delmas
Tomio Echigo
How-lung Eng
Boris Escalante
Chih-Peng Fan
Chiung-Yao Fang
Markus Flierl
Uwe Franke
Chiou-Shann Fuh
Zhi-Wei Gao
Richard Green
Nobuhara Hajime
Chin-Chuan Han
Hsueh-Ming Hang
Yutaka Hatakeyama
MahmoudReza Hejazi
Nancy Hitschfeld
Wenjen Ho
Jin Woo Hong
Hidekata Hontani
Yea-Shuan Huang
Hsu-Feng Hsiao
JunWei Hsieh
Chiou-Ting Hsu
Spencer Y.Hsu
Naoyuki Ichimura
Norikazu Ikoma
Atsushi Imiya
Herbert Jahn
Byeungwoo Jeon
Hong Jeong
Xiaoyi Jiang
Jesse Jin
Youngki Jung
Ramakrishna Kakarala
Yasushi Kanazawa
Li Wei Kang
Kazuhiko Kawamoto
Hideaki Kawano
Yukiko Kenmochi
Chnag-Ik Kim
Dongsik Kim
Hae-Kwang Kim
Hyoung Joong Kim
Jae-Gon Kim
Jin Woong Kim
Munchurl Kim
Sang-Kyun Kim
Yong Han Kim
Yoon Ho Kim
Ron Kimmel
Nahum Kiryati
Reinhard Koch
Andreas Koschan

Ki Ryong Kwon
Shang-Hong Lai
Ngoc-Minh Le
Chang-Ming Lee
Charng-Long Lee
Jia-Hong Lee
Pei-Jun Lee
Sangyoun Lee
Yung-Lyul Lee
Shipeng Li
Mark Liao
Wen-Nung Lie
Cheng-Chang Lien
Jenn-Jier James Lien
Vincenzo Liguore
Chia-Wen Lin
Chin-Teng Lin
Guo-Shiang Lin
Huei-Yung Lin
Shin-Feng Lin
Tom Lin
Damon Shing-Min Liu
Yuante Liu
Chun Shien Lu
Meng-Ting Lu
Yan Lu
Bruce MacDonald
Brendan McCane
Domingo Mery
Shaou-Gang Miaou
Andrew Vande Moere
Kyung Ae Moon
Ken'ichi Morooka
John Morris
Akira Nakamura
Philip Ogunbona
Seung-Jun Oh
Shigeru Owada
Jeng-Shyang Pan
Hyun Wook Park
Jong-II Park
Shmuel Peleg
Nicolai Petkov
Luis Pizarro
Richard Radke
Fernando Rannou
Ralf Reulke
Volker Rodehorst
Bodo Rosenhahn
Luis Rueda
Gary Sullivan
Hideo Saito
Tomoya Sakai
Day-Fann Shen
Jau-Ling Shih
Nobutaka Shimada
Donggyu Sim
Kwanghoon Sohn
Hwang-Jun Song
Gerd Stanke
Mu-Chun Su
Po-Chyi Su
Akihiro Sugimoto
Jae-Won Suh
Sang Hoon Sull
Hung-Min Sun
Yung-Nien Sun
Aramvith Supavadee
Alexis M. Tourapis
Toru Tamaki
Ewe Hong Tat
Matthias Teschner
C.-J. Tsai
Cheng-Fa Tsai
Chia-Ling Tsai
Y. Tim Tsai
Chien-Cheng Tseng
Din-Chang Tseng
Robert Valkenburg
Chieh-Chih Wang
Sheng-Jyh Wang
Wen-Hao Wang
Yuan-Kai Wang
Michael Wilkinson
Chee Sun Won
Jason Wood
Feng Wu
Hsien-Huang P. Wu
Shuicheng Yan
Jar-Ferr Yang
Mau-Tsuen Yang
Su Yang
Zhi-Fang Yang
Wei Yun Yau
Chia-Hung Yeh
Takanori Yokoyama
Lu Yu
Sung-Nien Yu
David Yuen
Huabei Zhou
Zhi Zhou

Sponsoring Institutions

National Tsing Hua University (NTHU)
National Chung Cheng University (NCCU)
The University of Auckland (UoA)
National Science Council (NSC)
Ministry of Education (MoE)
Sunplus Technology Co.
National Center for High-Performance Computing (NCHC)
Institute for Information Industry (III)

Table of Contents

3D Scene Modeling

Image Analysis

Intelligent Vision Applications

Multimedia Compression and Transmission

Multimedia Signal Processing

Panoramic Imaging and Distributed Video Systems

Sensor Technologies

Visualization

Error Analysis of Feature Based Disparity Estimation

Patrick A. Mikulastik, Hellward Broszio, Thorsten Thormählen, and Onay Urfalioglu

Information Technology Laboratory, University of Hannover, Germany
{mikulast, broszio, thormae, urfaliog}@lfi.uni-hannover.de
http://www.lfi.uni-hannover.de

Abstract. For real-time disparity estimation from stereo images the coordinates of feature points are evaluated. This paper analyses the influence of camera noise on the accuracy of feature point coordinates of a feature point detector similar to the *Harris Detector*, modified for disparity estimation. As a result the error variance of the horizontal coordinate of each feature point and the variance of each corresponding disparity value is calculated as a function of the image noise and the local intensity distribution. Disparities with insufficient accuracy can be discarded in order to ensure a given accuracy. The results of the error analysis are confirmed by experimental results.

1 Introduction

Disparity estimation algorithms compute disparities from the coordinates of selected corresponding feature points from images in standard stereo geometry. For the use of these estimated disparities in computer vision systems it is desirable to specify their accuracy. Therefore, in this paper the error variance of a disparity estimator is determined analytically and experimentally.

In previous work Luxen [1] measures the variance of feature point coordinates, taking image noise into account. The result is a mean error variance of all feature points in an image at a specific level of image noise. Local intensity distributions at specific feature points are not taken into account. Rohr [2] introduces an analytical model of a corner and calculates the feature point coordinates of different feature detectors for this corner. Thus he characterizes the different detectors but does not consider the errors of the coordinates due to camera noise. Szeliski [3] has analytically calculated the accuracy of displacement estimators like the KLT-Tracker [4]. The resulting covariance matrix describes the variance of the displacement error for each displacement. Other approaches do not estimate displacements but nevertheless apply similar covariance matrices to describe the accuracy of feature point coordinates [5,6,7,8]. However, these results have not been analytically proven or evaluated in experiments. Thus, so far the accuracy of feature point coordinates from image gradient based detectors similar to the *Harris Detector* [9] has not been calculated analytically.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1–12, 2006.

This paper analyses the influence of camera noise on the accuracy of a feature point detector for disparity estimation. It is based on a modified *Harris Detector* [9]. The accuracy is defined by the error variance of the feature coordinate. In a second step the error variance of the disparity estimation is derived.

In section 2 the feature detector considered in this paper is described. In section 3 an error analysis for feature coordinates and disparity is presented. Section 4 describes experiments for measuring the error variance of feature coordinates. Conclusions are given in section 5.

2 Feature Point Detector

The feature detector considered in this paper is based on the *Harris Detector* [9]. In stereo vision only disparities in horizontal direction of the stereo image are considered. Therefore, the process of feature detection is simplified so that only gradients in direction of the x-axis are measured. This also results in a reduced computational complexity.

For detection of feature points the following equation describes the edge response function R for vertical edges:

$$R(x,y) = \left| \sum_{i=-1}^{1} I_x(x, y+i)\alpha_i \right|, \qquad \alpha_i = [1,2,1] \tag{1}$$

where x and y are coordinates in the image. I_x is an approximation of the horizontal intensity gradient:

$$I_x(x,y) = -I(x-2,y) - 2I(x-1,y) + 2I(x+1,y) + I(x+2,y) \tag{2}$$

A feature point is detected, if $R(x,y)$ is greater than a predefined threshold T_R and if $R(x_m, y_m)$ is a local maximum in horizontal direction:

$$\begin{aligned} R(x_m,y_m) &> T_R \\ R(x_m,y_m) &> R(x_m-1,y_m) \\ R(x_m,y_m) &> R(x_m+1,y_m) \end{aligned} \tag{3}$$

Estimation of subpel coordinates. In horizontal direction a subpel coordinate is estimated for every feature point. A parabola is fitted to the edge response function (see figure 1):

$$R(x,y) = a + bx + \frac{1}{2}cx^2 \tag{4}$$

To achieve a compact notation the coordinate system is chosen so that $x_m = 0$. The three parameters a, b, c are calculated with:

$$\begin{aligned} R(-1,y_m) &= a - b + \frac{1}{2}c \\ R(0,y_m) &= a \\ R(+1,y_m) &= a + b + \frac{1}{2}c \end{aligned} \tag{5}$$

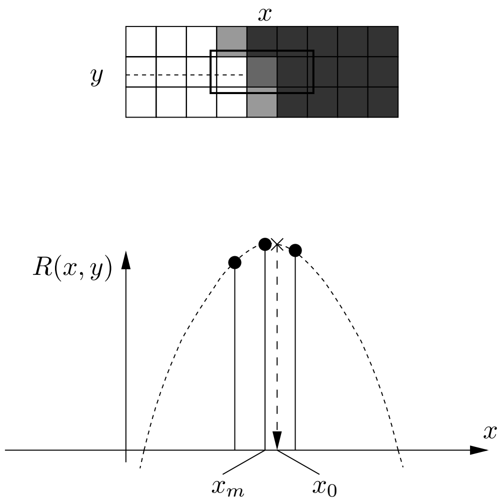


Fig. 1. Interpolation of $R(x, y)$ with a parabola. The maximum defines the subpel coordinate of the feature point x_0.

Solved for a, b, c:

$$
\begin{aligned}
a &= R(0, y_m) \\
b &= \frac{1}{2}\left(R(+1, y_m) - R(-1, y_m)\right) \\
c &= R(-1, y_m) - 2R(0, y_m) + R(+1, y_m)
\end{aligned}
\tag{6}
$$

In order to find the maximum of the parabola the derivation of equation 4 is set to zero:

$$
\frac{\partial R(x, y)}{\partial x} = b + cx \overset{!}{=} 0 \tag{7}
$$

The null of equation 7 marks the subpel coordinate x_0 of the feature point:

$$
\begin{aligned}
x_0 &= -\frac{b}{c} \\
&= \frac{1}{2}\frac{R(-1, y_m) - R(+1, y_m)}{R(-1, y_m) - 2R(0, y_m) + R(+1, y_m)}
\end{aligned}
\tag{8}
$$

3 Variance of the Horizontal Coordinate

To consider the influence of image noise on the subpel coordinates of feature points, noisy intensity values $\tilde{I}(x, y)$ with:

$$
\tilde{I}(x, y) = I(x, y) + n(x, y) \tag{9}
$$

are considered. $n(x, y)$ is white, mean free and Gaussian distributed noise with a noise variance of σ_n^2. The gradient of the image noise is defined as:

$$
n_x(x, y) = -n(x-2, y) - 2n(x-1, y) + 2n(x+1, y) + n(x+2, y) \tag{10}
$$

Therefore, the measured image gradient is $\tilde{I_x}(x,y)$:

$$\begin{aligned}\tilde{I_x}(x,y) &= -\tilde{I}(x-2,y) - 2\tilde{I}(x-1,y) + 2\tilde{I}(x+1,y) + \tilde{I}(x+2,y) \\ &= I_x(x,y) \underbrace{-n(x-2,y) - 2n(x-1,y) + 2n(x+1,y) + n(x+2,y)}_{n_x(x,y)} \\ &= I_x(x,y) + n_x(x,y)\end{aligned} \tag{11}$$

The measured cornerness response function $\tilde{R}(x,y)$ can be written as:

$$\begin{aligned}\tilde{R}(x,y) &= \left|\sum_{i=-1}^{1} \tilde{I_x}(x,y+i)\alpha_i\right| \\ &= \left|\sum_{i=-1}^{1} I_x(x,y+i)\alpha_i + \sum_{i=-1}^{1} n_x(x,y+i)\alpha_i\right|\end{aligned} \tag{12}$$

I_x can be positive or negative if the edge has a gradient in the positive or in the negative direction. The influence of the image noise is the same in both cases. Therefore only edges with positive gradients are considered in the following calculations and it is assumed that I_x is always positive:

$$\sum_{i=-1}^{1} I_x(x,y+i)\alpha_i > 0 \tag{13}$$

Generally, the intensity values are much larger than the image noise:

$$\sum_{i=-1}^{1} I_x(x,y+i)\alpha_i > \sum_{i=-1}^{1} n_x(x,y+i)\alpha_i \tag{14}$$

With equations 13 and 14 $\tilde{R}(x,y)$ can be written as:

$$\tilde{R}(x,y) = \underbrace{\left|\sum_{i=-1}^{1} I_x(x,y+i)\alpha_i\right|}_{R(x,y)} + \underbrace{\sum_{i=-1}^{1} n_x(x,y+i)\alpha_i}_{R_n(x,y)} \tag{15}$$

with:

$$R_n(x,y) = \sum_{i=-1}^{1} n_x(x,y+i)\alpha_i \tag{16}$$

$\tilde{R}(x,y)$ is computed for every point-position. The calculation of the feature point's subpel coordinates is carried out according to equation 3 to 7. $\tilde{x_0}$ can be calculated with equation 8:

$$\tilde{x_0} = \frac{1}{2}\frac{\tilde{R}(-1,y_m) - \tilde{R}(1,y_m)}{\tilde{R}(-1,y_m) - 2\tilde{R}(0,y_m) + \tilde{R}(1,y_m)} \tag{17}$$

and:

$$\tilde{x}_0 = \frac{1}{2}\frac{\tilde{R}(-1,y_m) - \tilde{R}(1,y_m)}{c + \underbrace{R_n(-1,y_m) - 2R_n(0,y_m) + R_n(1,y_m)}_{\Delta c}} \tag{18}$$

For a more compact notation Δc is introduced:

$$\Delta c = R_n(-1,y_m) - 2R_n(0,y_m) + R_n(1,y_m) \tag{19}$$

as well as the normalized value $\Delta c'$:

$$\Delta c' = \frac{\Delta c}{c} \tag{20}$$

c is a sum of intensity values and Δc is a sum of noise values. Therefore $\Delta c'$ is a small value. With Δc and $\Delta c'$ equation 18 simplifies to:

$$\begin{aligned} \tilde{x}_0 &= \frac{1}{2}\frac{\tilde{R}(-1,y_m) - \tilde{R}(1,y_m)}{c + \Delta c} \\ &= \frac{1}{2}\frac{\tilde{R}(-1,y_m) - \tilde{R}(1,y_m)}{c(1 + \Delta c')} \end{aligned} \tag{21}$$

Multiplication of nominator and denominator with $(1 - \Delta c')$ equals to:

$$\tilde{x}_0 = \frac{1}{2}\frac{\tilde{R}(-1,y_m) - \tilde{R}(1,y_m)}{c(1 - {\Delta c'}^2)} - \frac{1}{2}\frac{\tilde{R}(-1,y_m) - \tilde{R}(1,y_m)}{c(1 - {\Delta c'}^2)} \cdot \Delta c' \tag{22}$$

Since $\Delta c'$ is a small value it can be assumed that

$$1 >> {\Delta c'}^2 \qquad . \tag{23}$$

With this assumption, equation 22 is simplified to:

$$\tilde{x}_0 \approx \frac{1}{2}\frac{\tilde{R}(-1,y_m) - \tilde{R}(1,y_m)}{c} - \frac{1}{2}\frac{\tilde{R}(-1,y_m) - \tilde{R}(1,y_m)}{c} \cdot \Delta c' \tag{24}$$

With equation 16:

$$\begin{aligned} \tilde{x}_0 \approx\ & \overbrace{\left(\frac{R(-1,y_m) - R(1,y_m)}{2c}\right)}^{x_0} + \overbrace{\left(\frac{R_n(-1,y_m) - R_n(1,y_m)}{2c}\right)}^{\Delta x_a} \\ & - \left(\frac{R(-1,y_m) - R(1,y_m)}{2c} + \frac{R_n(-1,y_m) - R_n(1,y_m)}{2c}\right) \cdot \Delta c' \end{aligned} \tag{25}$$

Δx_a is defined:

$$\Delta x_a = \left(\frac{R_n(-1,y_m) - R_n(1,y_m)}{2c}\right) \tag{26}$$

With Δx_a we can write:

$$\tilde{x}_0 = x_0 + \Delta x_a - x_0 \Delta c' - \Delta x_a \Delta c' \tag{27}$$

$\Delta x_a \Delta c'$ is the procduct of sums of noise values. Therefore it is very small. It will be neglected in the following calculation:

$$\tilde{x}_0 = x_0 + \underbrace{\Delta x_a - x_0 \Delta c'}_{\Delta x_0} \tag{28}$$

Δx_0 describes the error of the undistorted coordinate x_0:

$$\Delta x_0 = \Delta x_a - x_0 \Delta c' \tag{29}$$

It has been verified that experimentally that Δx_0 has a zero mean. With this assumption the variance of the coordinate's error σ_Δ^2 equals the root mean square $E[\Delta x_0^2]$:

$$\begin{aligned} \sigma_\Delta^2 = E[\Delta x_0^2] &= E\left[(\Delta x_a - x_0 \Delta c')^2\right] \\ &= E\left[{\Delta x_a}^2 - 2x_0 \Delta x_a \Delta c' + x_0^2 {\Delta c'}^2\right] \\ &= E\left[{\Delta x_a}^2\right] - E\left[2x_0 \Delta x_a \Delta c'\right] + E\left[x_0^2 {\Delta c'}^2\right] \end{aligned} \tag{30}$$

The terms of equation 30 are evaluated individually:

$$\begin{aligned} E\left[{\Delta x_a}^2\right] &= E\left[\left(\frac{R_n(-1, y_m) - R_n(1, y_m)}{2c}\right)^2\right] \\ &= \frac{1}{4c^2} E\left[(R_n(-1, y_m) - R_n(1, y_m))^2\right] \end{aligned} \tag{31}$$

With equation 15:

$$E[{\Delta x_a}^2] = \frac{1}{4c^2} E\left[\left(\sum_{i=-1}^{1} n_x(-1, y_m + i)\alpha_i - \sum_{i=-1}^{1} n_x(1, y_m + i)\right)^2\right]$$

Evaluation of the square gives:

$$\begin{aligned} E[{\Delta x_a}^2] = \frac{1}{4c^2} E\Bigg[&\left(\sum_{i=-1}^{1} n_x(-1, y_m + i)\alpha_i\right)^2 - 2\sum_{i=-1}^{1} n_x(-1, y_m + i)\alpha_i \cdot \\ &\sum_{i=-1}^{1} n_x(1, y_m + i)\alpha_i + \left(\sum_{i=-1}^{1} n_x(1, y_m + i)\alpha_i\right)^2 \Bigg] \end{aligned} \tag{32}$$

With $E[n(x,y_1)n(x,y_2)]=0$, for $y_1 \neq y_2$ and $E[n(x_1,y)n(x_2,y)]=0$, for $x_1 \neq x_2$:

$$\begin{aligned} E&\left[\left(\sum_{i=-1}^{1} n_x(x,y+i)\alpha_i\right)^2\right] \\ &= E\left[\sum_{i=-1}^{1} n_x^2(x,y+i)\alpha_i^2\right] \\ &= E\Bigg[\sum_{i=-1}^{1} \big(n^2(x-2,y+i)+4n^2(x-1,y+i)+4n^2(x+1,y+i) \\ &\qquad\qquad +n^2(x+2,y+i)\big)\,\alpha_i^2\Bigg] \end{aligned} \tag{33}$$

and

$$\begin{aligned} E&\left[\sum_{i=-1}^{1} n_x(-1,y+i)\alpha_i \cdot \sum_{i=-1}^{1} n_x(1,y+i)\alpha_i\right] \\ &= E\left[\sum_{i=-1}^{1} n_x(-1,y+i)\cdot n_x(1,y+i)\alpha_i^2\right] \\ &= E\left[\sum_{i=-1}^{1} -4n^2(0,y+i)\alpha_i^2\right]. \end{aligned} \tag{34}$$

Equation 32 simplifies to:

$$\begin{aligned} E[\mathit{\Delta} {x_a}^2] = \frac{1}{4c^2} E\Bigg[&\sum_{i=-1}^{1}(n^2(-3,y_m+i)+4n^2(-2,y_m+i)+4n^2(0,y_m+i) \\ &+n^2(1,y_m+i))\alpha_i^2 - 2\sum_{i=-1}^{1} -4n^2(0,y_m+i)\alpha_i^2 \\ &+\sum_{i=-1}^{1}\big(n^2(-1,y+i)+4n^2(0,y+i)+4n^2(2,y+i) \\ &+n^2(3,y+i)\big)\,\alpha_i^2\Bigg] \end{aligned} \tag{35}$$

For the expectation the terms $n^2(x,y)$ become the variance of the image noise σ_n^2:

$$E[\mathit{\Delta} {x_a}^2] = \frac{1}{4c^2}\left[\sum_{i=-1}^{1} 10\sigma_n^2\alpha_i^2 - 2\left(\sum_{i=-1}^{1} -4\sigma_n^2\alpha_i^2\right) + \sum_{i=-1}^{1} 10\sigma_n^2\alpha_i^2\right] \tag{36}$$

Evaluation of the sums gives:

$$E[\mathit{\Delta} {x_a}^2] = \frac{\sigma_n^2}{4c^2}\left[60+48+60\right] = \frac{42\sigma_n^2}{c^2} \tag{37}$$

The second term in equation 30 can be expanded to:

$$\begin{aligned} E[2\Delta x_a x_0 \Delta c'] &= E\left[2\left(\frac{R_n(-1,y_m)-R_n(1,y_m)}{2c}\right)x_0\frac{\Delta c}{c}\right] \\ &= E\left[2\left(\frac{R_n(-1,y_m)-R_n(1,y_m)}{2c}\right)\right. \\ &\qquad \left. x_0\left(\frac{R_n(-1,y_m)-2R_n(0,y_m)+R_n(1,y_m)}{c}\right)\right] \\ &= \frac{x_0}{c^2}E\left[(R_n(-1,y_m)-R_n(1,y_m))\right. \\ &\qquad \left.(R_n(-1,y_m)-2R_n(0,y_m)+R_n(1,y_m))\right] \end{aligned} \tag{38}$$

A calculation similar to that from equation 31 to 36 leads to:

$$\begin{aligned} E[2\Delta x_a x_0 \Delta c'] &= \frac{x_0}{c^2}\left[\sum_{i=-1}^{1}10\sigma_n^2\alpha_i^2-\sum_{i=-1}^{1}10\sigma_n^2\alpha_i^2\right]\cdot \\ &\qquad \left[-2\left(\sum_{i=-1}^{1}4\sigma_n^2\alpha_i^2\right)+2\left(\sum_{i=-1}^{1}4\sigma_n^2\alpha_i^2\right)\right] \\ &= 0 \end{aligned} \tag{39}$$

The third term in equation 30 can be expanded to:

$$E\left[x_0^2\Delta c'^2\right] = \frac{x_0^2}{c^2}E\left[(R_n(-1,y_m)-2R_n(0,y_m)+R_n(1,y_m))^2\right] \tag{40}$$

Once again, a calculation similar to that from equation 31 to 36 leads to:

$$E\left[x_0^2\Delta c'^2\right] = \frac{120\sigma_n^2x_0^2}{c^2} \tag{41}$$

Insertion of equations 37, 39 and 41 in equation 30 leads to:

$$\begin{aligned} \sigma_\Delta^2 &= \frac{42\sigma_n^2}{c^2}+\frac{120\sigma_n^2x_0^2}{c^2} \\ &= \sigma_n^2\frac{42+120x_0^2}{c^2} \\ &= \sigma_n^2\frac{42+120x_0^2}{(R(-1,y_m)-2R(0,y_m)+R(+1,y_m))^2} \end{aligned} \tag{42}$$

This equation will be used to calculate the error variance σ_Δ^2 of the feature point coordinate. The disparity d is the distance between a feature point in the left image and the corresponding feature point in the right image:

$$d = x_{\text{left}} - x_{\text{right}} \tag{43}$$

The variance $\sigma^2_{\Delta_d}$ of the disparity error Δd is:

$$\begin{aligned} E\left[\Delta d^2\right] = \sigma^2_{\Delta_d} &= E\left[\left(\Delta x_{\text{left}} - \Delta x_{\text{right}}\right)^2\right] \\ &= E\left[\Delta x^2_{\text{left}} - 2\Delta x_{\text{left}}\Delta x_{\text{right}} + \Delta x^2_{\text{right}}\right] \\ &= E\left[\Delta x^2_{\text{left}}\right] - E\left[2\Delta x_{\text{left}}\Delta x_{\text{right}}\right] + E\left[\Delta x^2_{\text{right}}\right] \end{aligned} \tag{44}$$

The distortions of the feature point's coordinates in the two images are statistically independent, therefore equation 44 simplifies to:

$$\begin{aligned} \sigma^2_{\Delta_d} &= E\left[\Delta x^2_{\text{left}}\right] + E\left[\Delta x^2_{\text{right}}\right] \\ &= \sigma^2_{\Delta\text{left}} + \sigma^2_{\Delta\text{right}} \end{aligned} \tag{45}$$

The variance of the disparity error is given by the sum of the error variances of the feature point coordinates.

4 Experimental Results

The following experiments have been carried out to evaluate the derivation from the preceding chapter 3. An image sequence consisting of a static scene with constant lighting is taken with a 3-Chip CCD camera (Sony DXC-D30WSP). In order to determine the size of σ^2_n the average intensity value at each pixel from 1000 frames is calculated to produce a noise free image for the sequence. By subtraction of the noise free image and the original images, difference images containing only the camera noise can be obtained. A camera noise variance of $\sigma^2_n = 4.8$ which equals a PSNR of 41.3 dB was measured for the sequence. Figure 2 shows an example of an image from the sequence.

Using the feature detector described in section 2 feature points are detected in every image of the sequence. Now, correspondences between the feature points in the sequence are established. A correspondence is given, if a feature point in one image is located at the coordinates x, y and in another image at the coordinates $x \pm \epsilon, y \pm \epsilon$, with $\epsilon \leq 0,5$ pel If a feature point has correspondences in all images of a sequence, the measured variance of its horizontal coordinate $\tilde{\sigma}^2_\Delta$ is calculated. This value can be compared with the results from the derivation in section 3.

Figure 3 shows the measured variances $\tilde{\sigma}^2_\Delta$ over the variances σ^2_Δ calculated as described in section 3. The figure shows that the measured variances $\tilde{\sigma}^2_\Delta$ have nearly the same values as the calculated ones. Therefore the calculation is confirmed by the experiment. Also the values lie in the same regions observed by other researchers [1].

A second experiment with a synthetic image shows the dependence between subpel position of a feature point and the error variance of its coordinate. The image shown in figure 4 is used for this experiment. A feature detection in this image results in feature points with subpel positions in the whole range $-0.5 < x_0 < 0.5$ because the edge in the image is slightly slanted.

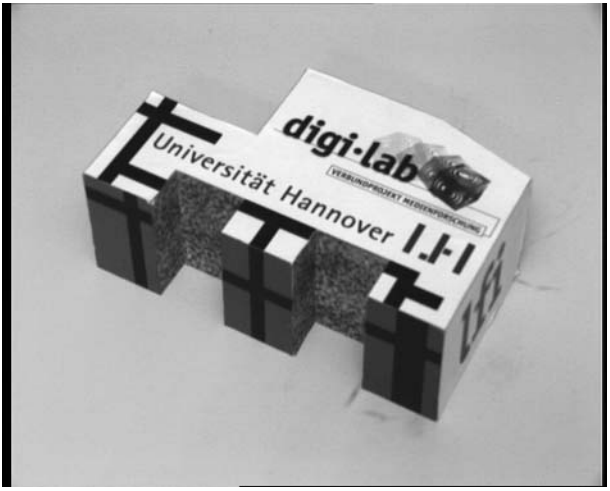


Fig. 2. Example image from the test sequence with a camera noise variance of $\sigma_n^2 = 4.8$

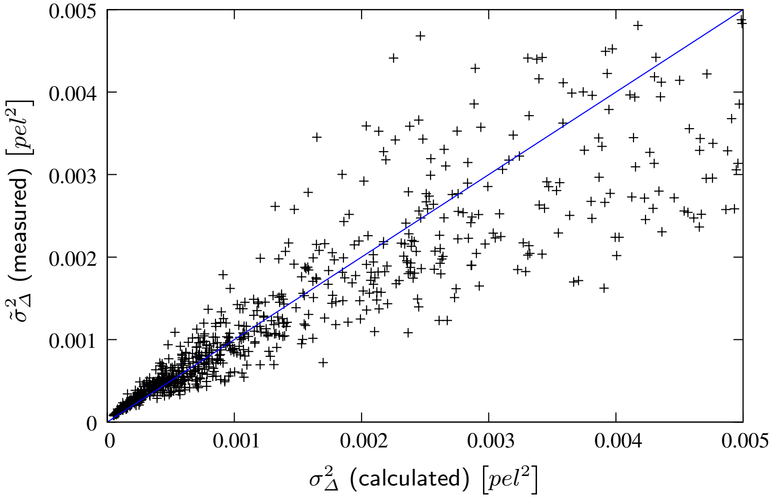


Fig. 3. Measured error variances $\tilde{\sigma}_\Delta^2$ over analytically calculated error variances σ_Δ^2 for each feature point in the image sequence taken with a real camera

Because the image shown in figure 4 is noisefree, synthetic noise was added the the images intensity values to generate 1000 noisy images with the original image as basis. Now the same procedure to calculate $\tilde{\sigma}_\Delta^2$ and σ_Δ^2 as described with the real image is conducted.

Figure 5 shows the measured and the calculated noise variances of the feature point's coordinates $\tilde{\sigma}_\Delta^2$ and σ_Δ^2 over the subpel coordinate. It can be observed that the coordinate error variance of feature points at a full-pel position is smaller

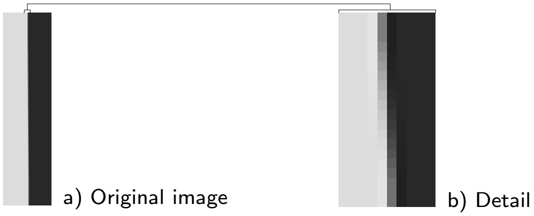


Fig. 4. Synthetic image used in the experiment. The edge is slightly slanted, so that feature points with a range of subpel coordinates can be detected.

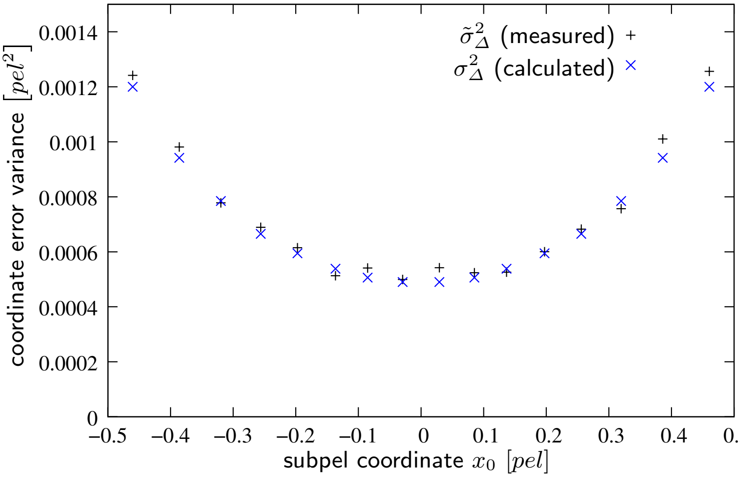


Fig. 5. Measured error variance $\tilde{\sigma}^2_\Delta$ and calculated error variance σ^2_Δ as function of the subpel coordinate x_0

than that for feature points at half-pel position. The variances vary by a factor of about three. Also it can be observed that the calculated variances σ^2_Δ match the measured variances $\tilde{\sigma}^2_\Delta$ which supports the correctness of the calculation.

5 Conclusions

A feature point detector using horizontal intensity gradients and offering subpel accuracy was described in section 2. It was shown that typically most of the feature points have an error variance of less than 0.01pel^2 for the horizontal coordinate. An analysis of the error of the horizontal feature point coordinate revealed the interrelationship between the image noise σ^2_n, the local image content, given by the local image intensity values $I(x, y)$, and the variance of the feature point's horizontal coordinate error σ^2_Δ. A formula for the disparity error variance based on the feature coordinate error variance has been derived. In an experiment (section 4) it was shown that the results of the analytical derivation match measured results obtained using synthetic images and images from a real camera. A second experiment has shown that the coordinate error variance of feature points at a full-pel position is smaller by a factor of three than that for feature points at half-pel position.

The calculation presented in this paper allows to benchmark feature points and disparities during feature detection on their expected error variance. This is a great advantage compared to methods that try to eliminate bad feature points at a later stage in the process of disparity estimation.

In the future this work will be expanded to a feature detector that measures gradients in all directions in the image, i.e. the feature detector of Harris et. al.[9]

References

1. Luxen, M.: Variance component estimation in performance characteristics applied to feature extraction procedures. In Michaelis, B., Krell, G., eds.: Pattern Recognition, 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003, Proceedings. Volume 2781 of Lecture Notes in Computer Science., Springer (2003) 498–506
2. Rohr, K.: Localization properties of direct corner detectors. Journal of Mathematical Imaging and Vision **4** (1994) 139–150
3. Szeliski, R.: Bayesian Modeling of Uncertainty in Low-Level Vision. Kluwer International Series in Engineering and Computer Science, 79, ISBN: 0792390393 (1989)
4. Shi, J., Tomasi, C.: Good features to track. In: CVPR, IEEE (1994) 593–600
5. Kanazawa, Y., Kanatani, K.: Do we really have to consider covariance matrices for image features? In: ICCV. (2001) 301–306
6. Morris, D.D., Kanade, T.: A unified factorization algorithm for points, line segments and planes with uncertainty models. In: Proceedings of Sixth IEEE International Conference on Computer Vision (ICCV'98). (1998) 696–702
7. Singh, A.: An estimation-theoretic framework for image-flow computation. In: ICCV. Third international conference on computer vision (1990) 168–177
8. Förstner, W.: Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision. In: CVGIP 40. (1987) 273–310
9. Harris, C., Stephens, M.: A combined corner and edge detector. In: 4th Alvey Vision Conference. (1988) 147–151

Collinearity and Coplanarity Constraints for Structure from Motion

Gang Liu[1], Reinhard Klette[1], and Bodo Rosenhahn[2]

[1] Department of Computer Science, The University of Auckland, New Zealand
[2] Max Planck Institute Saarbrücken, Germany

Abstract. Structure from motion (SfM) comprises techniques for estimating 3D structures from uncalibrated 2D image sequences. This work focuses on two contributions: Firstly, a stability analysis is performed and the error propagation of image noise is studied. Secondly, to stabilize SfM, we present two optimization schemes by using a priori knowledge about collinearity or coplanarity of feature points in the scene.

1 Introduction

Structure from motion (SfM) is an ongoing research topic in computer vision and photogrammetry, which has a number of applications in different areas, such as e-commerce, real estate, games and special effects. It aims at recovering 3D (shape) models of (usually rigid) objects from an (uncalibrated) sequence (or set) of 2D images.

The original approach [5] of SfM consists of the following steps: (1) extract corresponding points from pairs of images, (2) compute the fundamental matrix, (3) specify the projection matrix, (4) generate a dense depth map, and (5) build a 3D model. A brief introduction of some of those steps will be presented in Section 2.

Errors are inevitable to every highly complex procedure depending on real-world data, and this also holds for SfM. To improve the stabilization of SfM, two optimizations are proposed using information from the 3D scene; see Section 3. Section 4 presents experimental results, and Section 5 concludes the paper with a brief summary.

2 Modules of SfM

This section gives a brief introduction for some of the SfM steps (and related algorithms). For extracting correspondent points, we recall a method proposed in [14]. Then, three methods for computing the fundamental matrix are briefly introduced. To specify a projection matrix from a fundamental matrix, we describe two common methods based on [3,4]. In this step we also use the knowledge of intrinsic camera parameters, which can be obtained through Tsai calibration [12]; this calibration is performed before or after taking the pictures for the used

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 13–23, 2006.

camera. It allows to specify the effective focal length f, the size factors k_u and k_v of CCD cells (for calculating the physical size of pixels), and the coordinates u_0 and v_0 of the principal point (i.e., center point) in the image plane.

2.1 Corresponding Points

We need at least seven pairs of corresponding points to determine the geometric relationship between two images, caused by viewing the same object from different view points. One way to extract those points from a pair of images is as follows [14]:

(i) extract candidate points by using the Harris corner detector [2], (ii) utilize a correlation technique to find matching pairs, and (iii) remove outliers by using a LMedS (i.e., least-median-of-squares) method.

Due to the poor performance of the Harris corner detector on specular objects, this method is normally not suitable.

2.2 Fundamental and Essential Matrix

A fundamental matrix F is an algebraic representation of epipolar geometry [13]. It can be calculated if we have at least seven correspondences (i.e., pairs of corresponding points), for example using linear methods (such as the *8-Point Algorithm* of [8]) or nonlinear methods (such as the *RANSAC Algorithm* of [1], or the *LMedS Algorithm* of [14]).

In the case of a linear method, the fundamental matrix is specified through solving an overdetermined system of linear equations utilizing the given correspondences. In the case of a nonlinear method, subsets (at least seven) of correspondences are chosen randomly and used to compute candidate fundamental matrices, and then the best is selected, which causes the smallest error for all the detected correspondences.

According to our experiments, linear methods have a more time efficient and provide reasonably good results for large (say more than 13) numbers of correspondences. Nonlinear methods are more time consuming, but less sensitive to noise, especially if correspondences also contain outliers.

For given intrinsic camera parameters K_1 and K_2, the Essential matrix E can be derived from F by computing

$$E = K_2^T F K_1$$

2.3 Projection Matrix

A projection matrix P can be expressed as follows:

$$P = K[R \mid -RT]$$

where K is a matrix of the intrinsic camera parameters, and R and T are the rotation matrix and translation vector (the extrinsic camera parameters). Since the intrinsic parameters are specified by calibration, relative rotation and translation can be successfully extracted from the fundamental matrix F. When recovering the projection matrices in reference to the first camera position, the projection matrix of the first camera position is given as $P_1 = K_1[I \mid 0]$, and the projection matrix of the second camera position is given as $P_2 = K_2[R \mid -RT]$.

The method proposed by Hartley and Zisserman for computing rotation matrix R and translation vector T (from the essential matrix E) is as follows [3]:

1. compute E by using $E = K_2^T F K_1$, where

$$K_i = \begin{pmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

   (note: $K_1 = K_2$ if we use the same camera at view points 1 and 2),
2. perform a singular value decomposition (SVD) of E by following the template $E = Udiag(1,1,0)V^T$,
3. compute R and T (for the second view point), where we have two options, namely

$$R_1 = UWV^T \qquad R_2 = UW^TV^T$$

$$T_1 = u_3 \qquad T_2 = -u_3$$

   where u_3 is the third column of U and

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Another method for computing R and T from E (also only using elementary matrix operations) is given in [4], which leads to almost identical results as the method by Hartley and Zisserman.

2.4 Dense Depth Map

At this point, the given correspondences allow only a few points to be reconstructed in 3D. A satisfactory 3D model of a pictured object requires a dense map of correspondences. The epipolar constraint (as calculated above) allows that correspondence search can be restricted to one-dimensional epipolar lines, it supports that images are at first rectified following the method in [10], and that correspondence matching is then done by searching along a corresponding scan line in the rectified image. We also require a recovered base line between both camera positions to calculate a dense depth map.

3 Optimization with Prior Knowledge

Since computations of fundamental and projection matrix are sensitive to noise, it is necessary to apply a method for reducing the effect of noise (to stabilize SfM). We utilize information about the given 3D scene, such as knowledge about collinearity or coplanarity.

3.1 Knowledge About Collinearity

It is not hard to detect collinear points on man-made objects, such as buildings or furniture. Assuming ideal central projection (i.e., no lens distortion or noise), then collinear points in object space are mapped onto one line in the image plane. We assume that lens distortions are small enough to be ignored. Linearizing points which are supposed to be collinear can then be seen as a way to remove noise.

Least-square line fitting (minimizing perpendicular offsets) is used to identify the approximating line for a set of "noisy collinear points". Assume that we have such a set of points $P = \{(x_i, y_i) | i = 1, \ldots, n\}$ which determines a line $l(\alpha, \beta, \gamma) = \alpha x + \beta y + \gamma$. The coefficients α, β and γ are calculated as follows [7]:

$$\alpha = \frac{\mu_{xy}}{\sqrt{\mu_{xy}^2 + (\lambda^* - \mu_{xx})^2}}$$

$$\beta = \frac{\lambda^* - \mu_{xx}}{\sqrt{\mu_{xy}^2 + (\lambda^* - \mu_{xx})^2}}$$

$$\gamma = -(\alpha\overline{x} + \beta\overline{y})$$

where

$$\lambda^* = \frac{1}{2}(\mu_{xx} + \mu_{yy} - \sqrt{(\mu_{xx} - \mu_{yy})^2 + 4\mu_{xy}})$$

$$\mu_{xx} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2, \quad \mu_{yy} = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2$$

$$\mu_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}), \quad \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \texttt{and} \quad \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

After specifying the line, the points' positions are modified through perpendicular projection onto the line.

3.2 Knowledge About Coplanarity

Coplanar points can be expected on rigid structures such as on walls or on a tabletop. For a set of points, all incident with the same plane, there is a 3 × 3 matrix H called *homography* which defines a perspective transform of those points into the image plane [11].

Homography. Consider we have an image sequence (generalizing the two-image situation from before) and p_{ki} is the projection of 3D point P_i into the kth image, i.e. P_i is related to p_{ki} as follows:

$$p_{ki} = \omega_{ki} K_k R_k (P_i - T_k) \tag{1}$$

where ω_{ki} is an unknown scale factor, K_k denotes the intrinsic matrix (for the used camera), and R_k and T_k are the rotation matrix and translation vector. Following Equation (1), P_i can be expressed as follows:

$$P_i = \omega_{ki}^{-1} R_k^{-1} K_k^{-1} p_{ki} + T_k \tag{2}$$

Similarly, for point p_{li} lying on the lth image, we have

$$P_i = \omega_{li}^{-1} R_l^{-1} K_l^{-1} p_{li} + T_l \tag{3}$$

From Equations (2) and (3), we get

$$p_{ki} = \omega_{ki} K_k R_k (\omega_{li}^{-1} R_l^{-1} K_l^{-1} p_{li} + T_l - T_k) \tag{4}$$

With $R_{kl} = R_k R_l^{-1}$ we define $H_{kl}^{\infty} = K_k R_{kl} K_l^{-1}$. We also have epipole $e_{kl} = K_k R_k (T_l - T_k)$. Equation (4) can then be simplified to

$$p_{ki} = \omega_{ki} \omega_{li}^{-1} (H_{kl}^{\infty} p_{li} + \omega_{li} e_{kl}) \tag{5}$$

H_{kl}^{∞} is what we call the homography which maps points at infinity ($\omega_{li} = 0$) from image l to image k. Consider a point P_i on plane $\widehat{n}^T P_i - d = 0$. Then, from Equation (3), we have

$$\widehat{n}^T P_i - d = \widehat{n}^T \omega_{li}^{-1} R_l^{-1} K_l^{-1} p_{li} + \widehat{n}^T T_l - d = 0$$

Then we have

$$\omega_{li} = \frac{\widehat{n}^T R_l^{-1} K_l^{-1} p_{li}}{d - \widehat{n}^T T_l}$$

what can be rewritten as follows:

$$\omega_{li} = d_l^{-1} \widehat{n}^T R_l^{-1} K_l^{-1} p_{li}$$

where $d_l^{-1} = d – \widehat{n}^T T_l$ is the distance from the camera center (principal point) of the lth image to the plane $(\widehat{n}, d)$. Substituting ω_{li} into Equation (5), finally we have

$$p_{ki} = \omega_{ki} \omega_{li}^{-1} (H_{kl}^{\infty} + d_l^{-1} e_{kl} \widehat{n}^T R_l^{-1} K_l^{-1}) p_{li}$$

Let

$$H = \omega_{ki} \omega_{li}^{-1} (H_{kl}^{\infty} + d_l^{-1} e_{kl} \widehat{n}^T R_l^{-1} K_l^{-1})$$

This means: points lying in the same plane have identical H which can be utilized as coplanarity constraint; see [11].

Coplanarity optimization. Coplanar points satisfy the relation described by homography. We use this relation for modifying "noisy coplanar points," using the following equation:

$$p_{ki} = H_{kl} p_{li}$$

Here, H_{kl} is the homography between kth and lth image in the sequence, and p_{ki}, p_{li} are projections of point P_i on the kth and lth image, respectively.

4 Experiments and Analysis

To analyze the influence of noise, we perform SfM in a way as shown in Figure 1. At Step 1, Gaussian noise is introduced into coordinates of detected correspondences. At step 2, three different methods are compared to specify which one is the best to compute the fundamental matrix. At Step 3, a quantitative error analysis is performed.

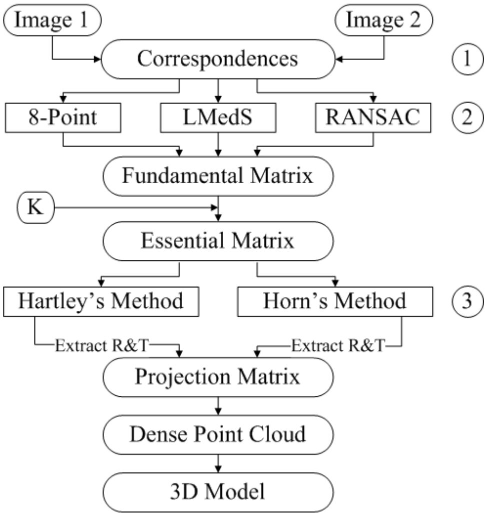


Fig. 1. The way we perform SfM

This section shows at first experiments of the performance of different methods for computing the fundamental matrix, and second the effect of those optimizations mentioned in the previous section.

4.1 Computation of Fundamental Matrix

Three algorithms (8-Point, RANSAC and LMedS) are compared with each other in this section. To specify the most stable one in presence of noise, Gaussian Noise (with mean 0 and deviation $\delta = 1$ pixel) and one outlier are propagated to given correspondences. Performances of the three algorithms are characterized in Figure 2: due to the outlier, the 8-Point Algorithm is more sensible than the other two.

4.2 Optimizations

To test the effect of the optimizations mentioned in the previous section, the results of splitting essential matrices (rotation matrices and translation vectors) are utilized to compare with each other. Two images of a calibration object are used as test images (shown in Figure 3). The data got from calibration (intrinsic

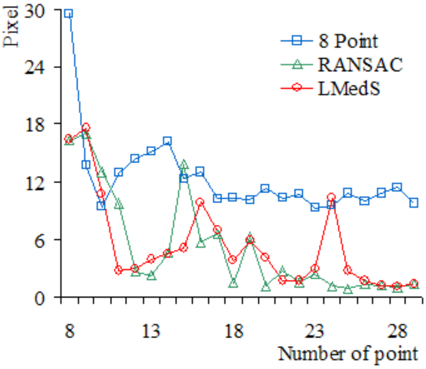


Fig. 2. Performance of three algorithms in presence of noise

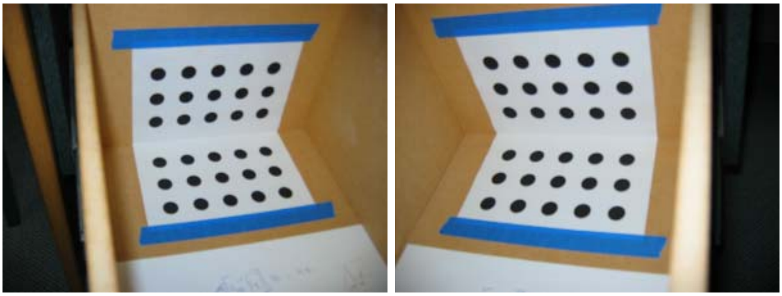


Fig. 3. The first (left) and second (right) candidate images

parameters and extrinsic parameters of camera) are used as the ground truth. Roll angle α, pitch angle β and yaw angle γ are used to compare the rotation matrices in a quantitative manner. These angles can be computed from a rotation matrix R by following equations [9]:

$$\begin{aligned} \alpha =& \ \ \text{atan2}(\tfrac{r_{23}}{sin(\gamma)}, \tfrac{r_{13}}{sin(\gamma)}) \\ \beta =& \ \ \text{atan2}(\tfrac{r_{32}}{sin(\gamma)}, \tfrac{-r_{31}}{sin(\gamma)}) \\ \gamma =& \ \text{atan2}(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}) \end{aligned}$$

where r_{ij} is the element of R at ith row and jth column, and

$$atan2(y,x) = \begin{cases} atan(\frac{y}{x}) & (x > 0) \\ \frac{y}{|y|} \cdot (\pi - atan(|\ \frac{y}{x}\ |)) & (x < 0) \\ \frac{y}{|y|} \cdot \frac{\pi}{2} & (y \neq 0, x = 0) \\ \text{undefined} & (y = 0, x = 0) \end{cases}$$

Since splitting the essential matrix only results in a translation vector up to a scale factor, all translation vectors (include the ground true one) are transformed into a normalized vector (length equal to one unit) to compare with each other

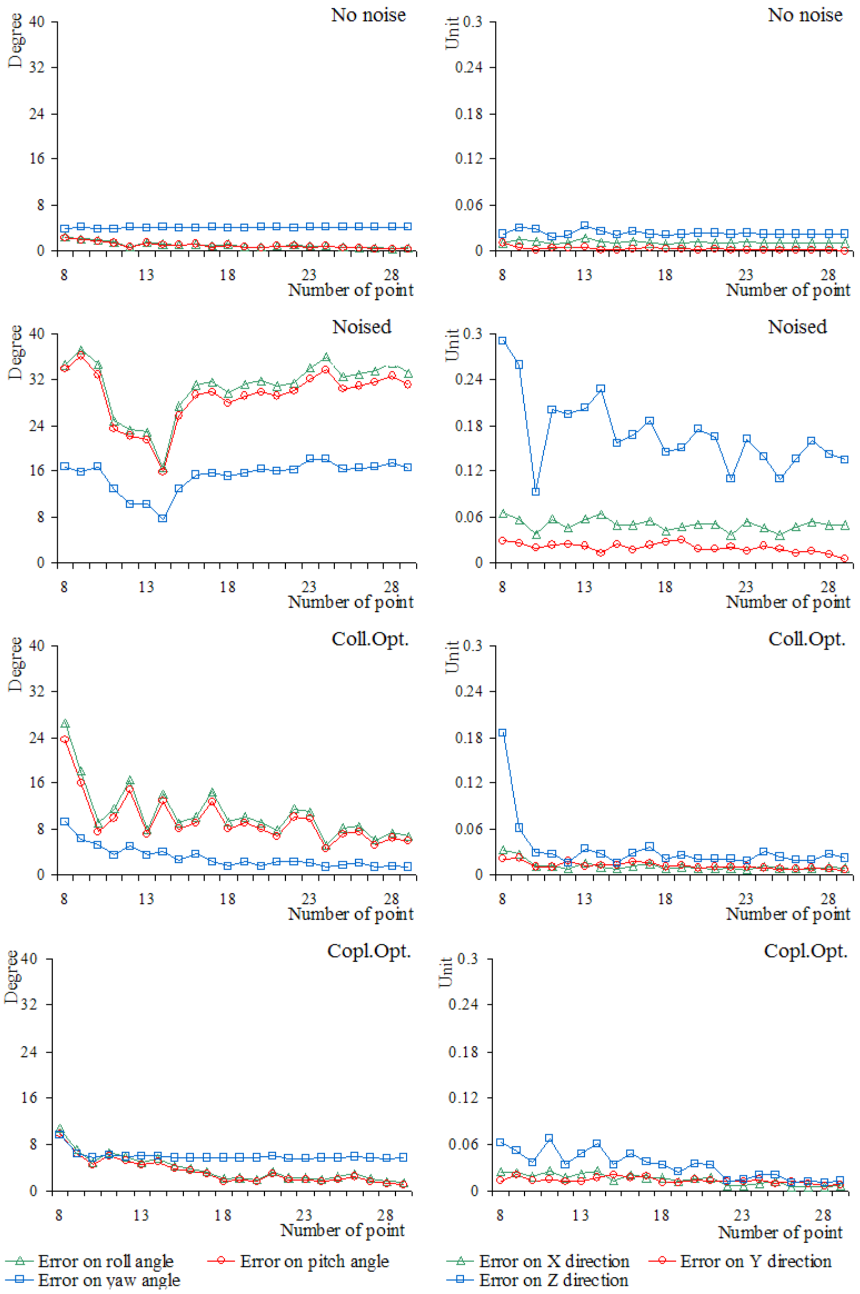


Fig. 4. Errors in rotation matrices (left) or translation vectors (right). First row: errors from non-noisy data. Second row: noisy data. Third or fourth row: errors from noisy data after optimization with collinearity or coplanarity knowledge, respectively.

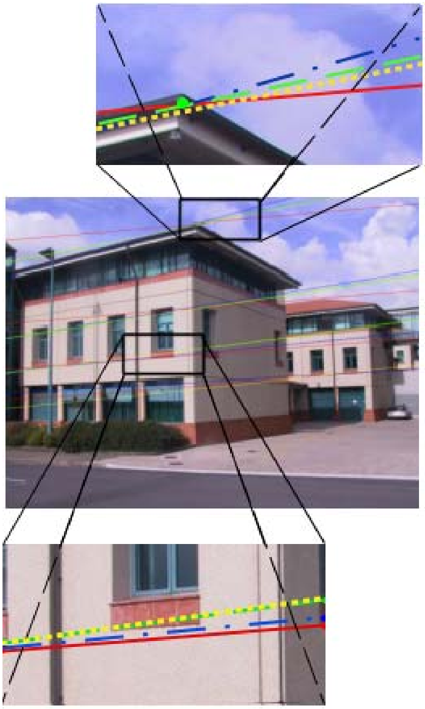


Fig. 5. Epipolar lines result from different data sets. The green lines (dash lines) from data without generated noise; the red lines (straight lines) are from noisy data; the blue lines (dash-dot lines) and yellow lines (dot lines) are from noisy data which has been optimized with collinearity and coplanarity knowledge, respectively.

in a quantitative manner. The comparison of rotation matrices and translation vectors are shown in Figure 4. The errors are mean error of ten times iteration when different number of correspondences are given. The noise propagated is Gaussian noise (with mean 0 and deviation $\delta = 1$ pixel). The method used to compute the fundamental matrix is the 8-Point Algorithm, which is more sensitive to noise than RANSAC and LMedS Algorithm. The method of Hartley and Zisserman is used to split essential matrix.

According to the results shown in Figure 4, the coplanarity knowledge gives a better optimization than collinearity knowledge. One possible reason is that the collinearity optimization is performed on uncalibrated images, in which the true correlation of collinear points are not strictly lying in a straight line.

For arbitrary images, the effect of optimizations can be seen from Figure 5 through looking at relative positions of epipolar lines computed from different data sets. It shows that the two optimization strategies bring positive effects on reducing the influence of noise, and the coplanarity optimization performs better than the collinearity optimization. Figure 6 shows the reconstructed point cloud of the CITR-building in Auckland and Figure 7 visualizes the texture mapped surface model. The main edges of the building are reconstructed with

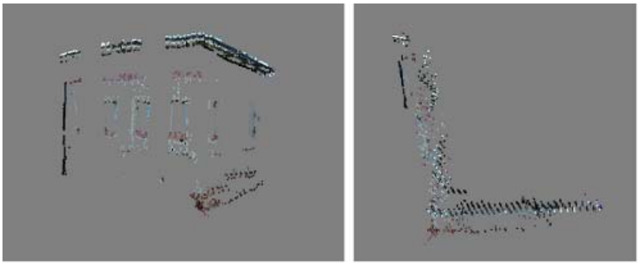


Fig. 6. Two different views of reconstructed points from the optimized SfM algorithm

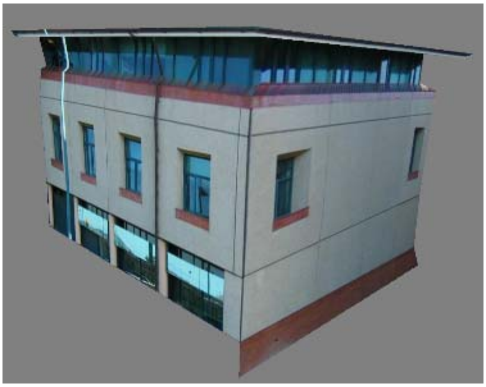


Fig. 7. Triangulated surface mesh with textures

near-perfect 90° angles. Slight image noise (less then 1 pixel) already leads to angles between 20° and 140° which indicates the sensitivity of classic SfM approaches. By incorporating the collinearity and coplanarity constraints, the reconstruction quality improved.

5 Conclusion

Modules relating to structure from motion have been discussed in this paper. According to experiments, structure from motion is sensitive to noise and it is necessary to improve its stability. Two optimizations, using collinearity and coplanarity knowledge, have been proposed, and the relating experiments show that the two proposed optimizations, especially the coplanarity one, bring positive effects on reducing influences of noise.

Acknowledgments

The authors thank Daniel Grest and Kevin Koeser from the University of Kiel for the BIAS-Library and useful hints.

References

1. M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, **24**:381–385, 1981.
2. C. Harris and M. Stephen. A combined corner and edge detector. In Proc. *Alvey Vision Conf.*, pages 147–151, 1988.
3. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, 2000.
4. B. K. P. Horn. Recovering baseline and orientation from essential matrix. http://www. ai.mit.edu/people/bkph/papers/essential.pdf, 1990.
5. T. Huang. Motion and structure from feature correspondences: a review. *Proc. IEEE*, **82**:252–268, 1994.
6. R. Klette, K. Schlüns, and A. Koschan. *Computer Vision – Three-dimensional Data from Images*. Springer, Singapore, 1998.
7. C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice Hall, Englewood, 1974.
8. H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, **293**:133–135, 1981.
9. R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, 1994.
10. M. Pollefeys, R. Koch, and L. van Gool. A simple and efficient rectification method for general motion. In Proc. *Int. Conf. Computer Vision*, pages 496–501, 1999.
11. R. Szeliski and P. H. S. Torr. Geometrically constrained structure from motion: points on planes. In Proc. *European Workshop 3D Structure Multiple Images Large-Scale Environments*, pages 171–186, 1998.
12. R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. Robotics and Automation*, **3**:323–344, 1987.
13. Z. Zhang. Determining the epipolar geometry and its uncertainty: a review. *Int. J. Computer Vision*, **27**:161–198, 1998.
14. Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence J.*, **78**:87–119, 1995.

Reconstruction of Building Models with Curvilinear Boundaries from Laser Scanner and Aerial Imagery

Liang-Chien Chen, Tee-Ann Teo, Chi-Heng Hsieh, and Jiann-Yeou Rau

Center for Space and Remote Sensing Research, National Central University
No. 300, Jhong-Da Road, Jhong-Li City, Tao-Yuan 32001, Taiwan
{lcchen, ann}@csrsr.ncu.edu.tw, 93322082@cc.ncu.edu.tw,
jyrau@csrsr.ncu.edu.tw

Abstract. This paper presents a scheme to detect building regions, followed by a reconstruction procedure. Airborne LIDAR data and aerial imagery are integrated in the proposed scheme. In light of the different buildings, we target the ones with straight and curvilinear boundaries. In the detection stage, a region-based segmentation and object-based classification are integrated. In the building reconstruction, we perform an edge detection to obtain the initial building lines from the rasterized LIDAR data. The accurate arcs and straight lines are then obtained in the image space. By employing the roof analysis, we determine the three dimensional building structure lines. Finally, the Split-Merge-Shape method is applied to generate the building models. Experimental results indicate that the success rate of the building detection reaches 91%. Among the successfully detected buildings, 90% of the buildings are fully or partially reconstructed. The planimetric accuracy of the building boundaries is better than 0.8m, while the shaping error of reconstructed roofs in height is 0.14 m.

Keywords: LIDAR, Aerial Image, Building Models.

1 Introduction

Building modeling in cyber space is an essential task in the application of three-dimensional geographic information systems (GIS) [1]. The extracted building models are useful for urban planning and management, disaster management, as well as other applications.

Traditionally, the generation of building models is mainly performed by using stereo aerial photography. However, the airborne LIDAR (Light Detecting And Ranging) system is proving to become a promising technological alternative. As the airborne LIDAR integrates the Laser Scanner, Global Positioning System (GPS) and Inertial Navigation System (INS) together, it is able to provide direct georeferencing. Its high precision in laser ranging and scanning orientation renders possible decimeter level accuracy of 3D objects. The three-dimensional point clouds acquired by an airborne LIDAR system provide comprehensive shape detail, while aerial images contain plentiful spectral information. Thus, the integration of the two complementary data sets reveals the possibility of an automatic generation of building models. Several data fusion methods have been proposed to generate building models, e.g.,

L.-W. Chang, W.-N. Lie, R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 24 – 33, 2006.

LIDAR and aerial images, [2] LIDAR and three-line-scanner images [3], LIDAR and satellite images [4], LIDAR, aerial images and 2D maps [5].

The physical mechanism in the model generation includes the identification of the buildings region, and the reconstruction of the geometric models. To automate the identification procedure, a classification process using remotely sensed data should be employed to detect the building regions. The reconstruction strategy can be classified into two categories, i.e., model-driven and data-driven. The Model-driven approach is a top-down strategy, which starts with a hypothesis of a synthetic building model. Verification of the model's consistency with the LIDAR point clouds is then performed. In the strategy, a number of 3D parametric primitives are generated by the segmentation of the LIDAR data. Afterwards, the best fitting primitives is selected from the aerial image. The building model is obtained by merging together all the 3D building primitives [6]. This method is restricted by the types of 3D parametric primitives. The Data-driven approach is a bottom-up strategy, which starts from the extractions of the building primitives, such as building corner, structure lines and roof-tops. Subsequently, a building model can be grouped together through a hypothesis process. A general approach is to extract the plane features from the LIDAR point clouds, and detect the line features from the aerial image. The plane and line features are combined to develop the building models [7]. The reported results are limited to buildings with straight line boundaries. Buildings with curvilinear boundaries are seldom discussed. Furthermore, there is no report in the literature on 3D curvilinear building modeling from LIDAR and image data.

From a data fusion's point of view, we propose a scheme to reconstruct building models via LIDAR point clouds and aerial image. The proposed scheme comprises of two major parts: (1) detection, and (2) reconstruction. Spatial registration of the LIDAR data and aerial imagery is performed during the data preprocessing. The registration is done in such a way that the two data sets are unified in the object coordinate system. Meanwhile, we calculate the exterior orientation parameters of the aerial imagery by employing ground control points. Afterwards, a region-based segmentation and object-based classification are integrated during the building detection stage. After the segmentation, the object-based classification method detects the building regions by considering the spectral features, shape, texture, and elevation information. For the building reconstruction stage, the building blocks are divided and conquered. Once the building regions are detected, we analyze the coplanarity of the LIDAR point clouds to obtain the 3D planes and 3D ridge lines. We use the edge detection method to obtain the initial building lines from the rasterized LIDAR data. Through the back projection of the initial lines to the image space, the accurate arcs and straight lines are obtained in the image space. The edges extracted from the aerial image are incorporated to determine the 3D position of the building structure lines. A patented Split-Merge-Shape [8] method is then employed to generate the building models in the last step.

This article is organized as follows. Section 2 discusses the methodology of the building detection. In section 3, the building reconstruction strategy is presented. We validate the proposed scheme by using aerial image and LIDAR data acquired by the Leica ALS50 system in section 4. Finally, a summary of the described method is given at the last segment.

2 Building Detection

The primary objective of this section is to extract the building regions. There are two steps in the proposed scheme: (1) region-based segmentation, and (2) object-based classification. The flow chart of the detection method is shown in Fig. 1.

There are two ways to conduct the segmentation. The first is the contour-based approach. It performs the segmentation by utilizing the edge information. The second is the region-based segmentation. It uses a region growing technique to merge pixels with similar attributes. We select the region-based approach, because it is less sensitive to noise. The proposed scheme combines the surface variations from the LIDAR data with the spectral information obtained from the orthoimage in the segmentation. The pixels with similar geometric and spectral properties are merged into a region.

After segmentation, each region is a candidate object for classification. Instead of a pixel-based approach, an object-based approach is performed. Considering the characteristics of elevation, spectral information, texture, roughness, and shape, the classification procedure is performed to detect the building regions. The considered characteristics are described as follows.

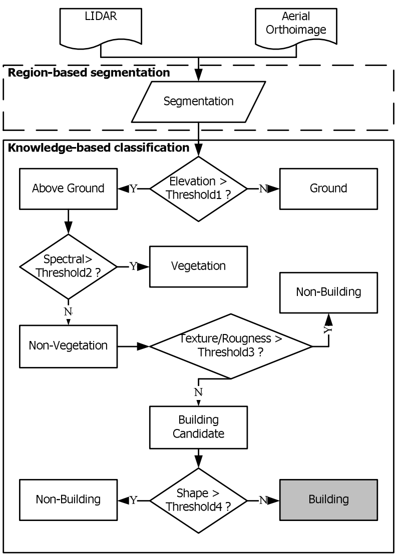


Fig. 1. Flowchart of building detection

(1) Elevation: Subtracting the Digital Terrain Model (DTM) from the Digital Surface Model (DSM), we generate the Normalized DSM (NDSM). The data describes the height variations above ground. By setting an elevation threshold, one can select the above ground objects, which include buildings and vegetation.
(2) Spectral information: The spectral information is obtained from color aerial image. A greenness index is used to distinguish the vegetation from non-vegetation areas.
(3) Texture: The texture information is retrieved from aerial image via a Grey Level Co-occurrence Matrix (GLCM) [9] texture analysis. GLCM is a matrix of relative frequencies for pixel values occurring within a specific neighborhood. We select the entropy and homogeneity as indices to quantify the co-occurrence probability. The role of the texture information is to separate the building from the vegetation, when the objects have similar spectral responses.
(4) Roughness: The roughness of the LIDAR data aims to differentiate the vegetation regions from non-vegetation ones. The surface roughness is similar to the texture information of the image data. The role of the surface roughness is to separate the building and vegetation, when the objects have similar spectral responses. We choose the slope variance as the roughness index.

(5) Shape: The shape attribute includes the size and length-to-width ratio. An area threshold is used to filter out the overly small objects. This means regions smaller than a minimum area are not taken into account as a building. The length-to-width ratio is suitable to remove the overly thin objects. The objects would not be considered as a building, when the length-to-width ratio is larger than a specified threshold.

3 Building Reconstruction

The reconstruction stage begins by isolating each individual building region. The stage includes three parts: (1) detection of roof planes, (2) extraction of 3D structure lines, and (3) 3D building modeling. The flow chart of the building reconstruction is shown in Fig. 2.

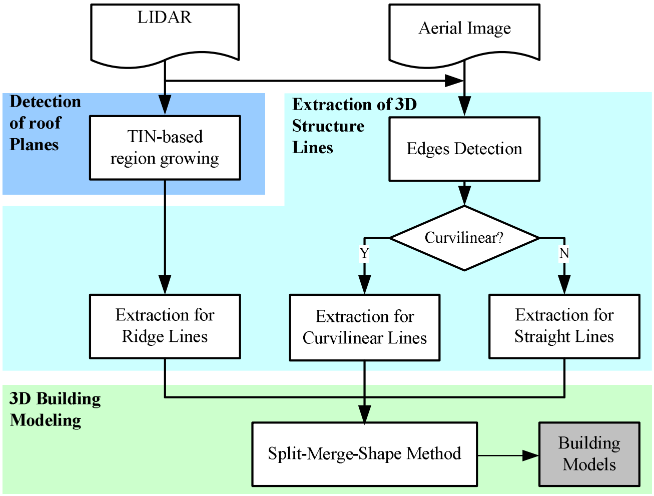


Fig. 2. Flowchart of building reconstruction

3.1 Detection of Roof Planes

A TIN-based region growing procedure is employed to detect the roof planes. The point clouds are first structured to a TIN-mesh built by Delaunay triangulation. The coplanarity and adjacency between the triangles are considered for the growing TIN-based regions. The coplanarity condition is examined by the distance of the triangle center to the plane. When the triangles meet the coplanarity criteria, the triangles are merged as a new facet. The process starts by selecting a seed triangle and determining the initial plane parameter. The initial plane is determined from the seed triangle. If the distance of the neighbor triangle to the initial plane is smaller than a specified threshold, the two triangles are combined. The parameters of the reference plane are recalculated using all of the triangles that belong to the region. The seed region starts to grow in this manner. When the region stops growing, a new seed triangle is chosen.

The region-growing stops when all of the triangles have been examined. Due to the errors of the LIDAR data, the detected regions may consist of fragmental triangles. Thus, small regions that have the closest normal vector will be merged in its neighborhood. After the region growing, we use the least squares regression to determine the plane equations. A sample result of the detection is illustrated in Fig. 3.

(a) (b)

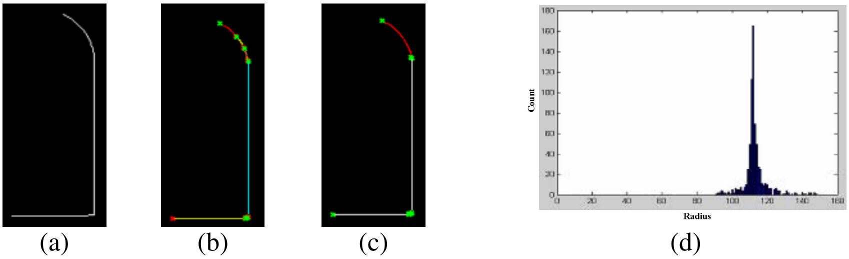
Fig. 3. Illustration of detection for roof planes (a) triangles mesh (b) extracted planes

3.2 Extraction of 3D Structure Lines

Two types of building structure lines, namely, ridge lines and step edges are targeted in this study. The ridge line is a building feature, where two planes intersect. It can be determined by the extracted planes. The step edge represents a building structure, where roofs have height jumps. A step edge may be straight or curvilinear. Considering the difference in the spatial resolution, each initial step edge is estimated from the LIDAR data, while the precise step edge is extracted from the image.

In the extraction of ridges, the line is obtained by the intersection of the neighboring planes. Mathematically, the intersection line computed from the plane equations is a 3D straight line without end points. Thus, we use the shared triangle vertices to define the line ends. That means the final product of a ridge line is a straight line with two end points.

In the extraction of step edges, we detect the initial building edges from the rasterized LIDAR data. The rough edges from the LIDAR data are used to estimate the location of the step edges in the image space. Building edges around the projected area are detected through the Canny Edge Detector [10]. At this stage, there is no pre-knowledge about the lines being straight or curvilinear. In order to distinguish the straight lines from the curvilinear ones, we develop a scheme to identify the different line types. The basic mechansim is to determine the most probable radius of a segment. First, we perform the line tracking to split all the edges into several line segments. A sample of the extracted edge pixels are shown in Fig. 4a. Fig. 4b demonstrates a sample result of the split line segments. Afterwards, we merge the adjacent line segments by the criterion of length and angle. The merged lines are treated as an arc candidate. Fig. 4c presents a sample result of the arc candidate. The last step is to test the rationality of the radius for each arc candidate. We randomly select three points from an arc candidate to calculate a radius. All the points are tested to generate a radius histogram like Fig. 4d. The horizontal axis is for the radii of possible circles, while the vertical axis presents the accumulated number of the radii. The arc candidate is accepted when the radius shows the highest concentration.



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

Fig. 4. Illustration of curvilinear lines separation (a) extracted edges (b) split line segments (c) arc candidates (d) radius histogram

For the classified straight lines, we use the Hough Transform [11] to extract the target lines in a parameter space. Eq. 1 shows a straight line being transformed in the Hough space. On the other hand, the classified curvilinear lines are extracted by a modified Hough Transform [12]. The circle equation is shown in Eq. 2. Notice that the circle's radius is calculated from the radius histogram, as described above. Given the image coordinates and the height information from the 3D planes, we calculate the 3D structure lines in the object space via exterior orientation parameters.

$$x_i \cos\theta + y_i \sin\theta = \rho \ . \tag{1}$$

where,

x_i,y_i: the pixel coordinate in location i,
θ: angle, and
ρ: distance.

$$(x_i - a)^2 + (y_i - b)^2 = r^2 \ . \tag{2}$$

where,

xi,yi: the pixel coordinate in location i,
a,b: the center of a circle, and
r: radius of circle.

3.3 3D Building Modeling

The extracted 3D structure lines are processed by a patented method, i.e., Split-Merge-Shape method [8], for building reconstruction. The *Split* and *Merge* process sequentially reconstructs the topology between the two consecutive line segments, and then reforms the areas as enclosed regions. The two procedures are performed in a two dimensional space. During splitting, a line segment is chosen for reference. We split all the line segments into a group of roof primitives. All of the possible roof primitives are generated by splitting the area of interest from all the line segments. In the merging procedure, the connectivity of the two adjacent roof primitives is analyzed successively. If the boundaries shared between them do not correspond to any 3D line segment, the two roof primitives will be merged.

The *Shape* step uses the available 3D edge height information to determine the most appropriate rooftop. The *Shape* process is performed in a three dimensional space. The first step of shaping is to assign a possible height for each roof edge from its corresponding 3D edge. Every 3D edge is first automatically labeled as a shared edge or an independent edge. The height for an independent edge can then be assigned from its corresponding 3D edges. The second step is to define the shape of a rooftop, according to the height of the independent edges. If more than two independent edges exist, and are sufficient to fit into a planar face, a coplanar fitting is applied. Fig. 5 demonstrates the modeling procedure.

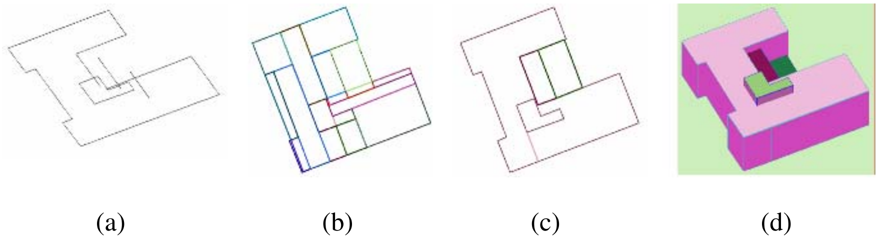


Fig. 5. Procedure of building modeling (a) 3D line segments (b) results of splitting (c) results of merging (d) results of shaping

4 Experimental Results

The LIDAR data used in this investigation covers a test area situated within the Industrial Technology Research Institute in northern Taiwan. The LIDAR data is obtained by the Leica ALS 50 system. The average density of the LIDAR point clouds is 2pts/m^2. The LIDAR data is shown in Fig 6a. The ground sampling distance of the aerial image is 0.5m. Fig 6b shows the image of the test area. The test area contains complex buildings, such as straight lines and curvilinear boundary buildings. The roof type is either flat or gable. There are 23 buildings in the test area.

We use stereoscopic measurements to derive the building models, as references for validations. The experiments include three different aspects in the validation procedure. The first evaluates the detection rate for building regions. The second checks the planimetric accuracy of building corners. The third assesses the height discrepancy between the roof top and the original LIDAR point clouds.

4.1 Building Detection

During building detection, the surface points and ground points from the LIDAR data are both rasterized to DSM and DTM with a pixel size of 0.5m. The aerial image is orthorectified by using the DSM. A 1/1,000 scale topographic map is employed for ground truth. The classified results, which are superimposed onto a topographic map is shown in Fig. 6c. It is found that 21 out of the 23 buildings are successfully detected, where the detection rate is 91%. The missing buildings are, in general, too small for detection. Both of the two missing buildings are smaller than 30 m^2.

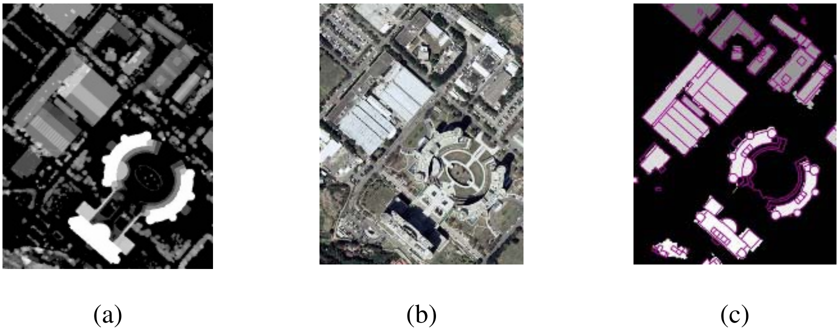


Fig. 6. Results of building detection (a) LIDAR DSM (b) aerial image (c) detected building regions with vector maps

4.2 Building Reconstruction

During building reconstruction, we categorize the buildings into three types: (1) straight line buildings with flat roofs, (2) straight line buildings with gable roofs, and (3) curvilinear boundary buildings with flat roofs. Two selected examples with different building complexities are given in Fig. 7. Fig. 8 shows all the generated building models.

In the accuracy evaluation, we compare the coordinates of the roof corners in the reconstructed models with the corners acquired by the stereoscopic manual measurements. The Root-Mean-Square-Errors (RMES) are 0.71m and 0.73m in the X and Y directions, respectively. The ground resolution of the aerial image is 0.5m. Thus, the accuracy is roughly 1.5 pixels in the image space. In Fig. 9, we provide error vectors that are superimposed onto the building boundaries.

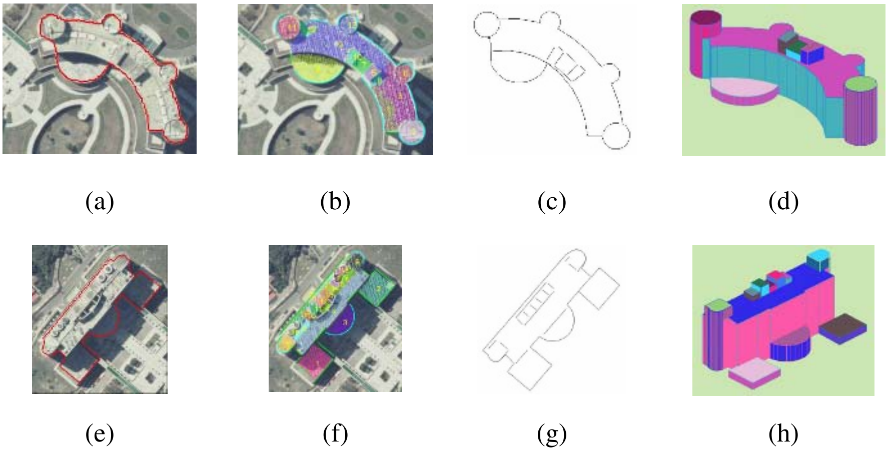


Fig. 7. Results of reconstruction for complex buildings (a) building regions of case 1, (b) detected planes of case 1, (c) extracted lines of case 1, (d) generated building models of case 1, (e) building regions of case 2, (f) detected planes of case 2, (g) extracted lines of case 2, (h) generated building models of case 2

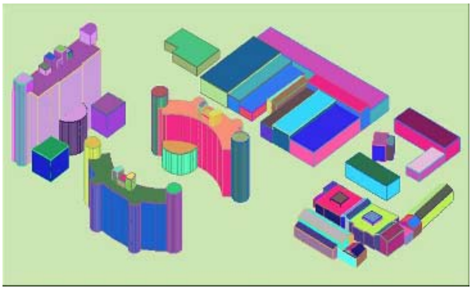


Fig. 8. The generated building models

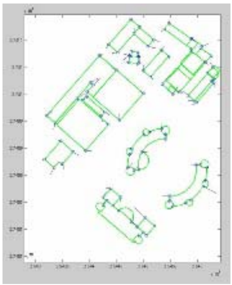


Fig. 9. Error vectors of building corners

Table 1. Success rate of building reconstruction

Reconstruction results	Straight line boundaries		Curvilinear boundaries	Success rate (%)
	Flat roof	Gable roof	Flat roof	
Correct	9	5	2	76
Partially correct	2	0	1	14
Erroneous	0	2	0	10

The fidelity of the building reconstruction is validated in terms of the success rate. The success rate of the reconstruction is divided into three categories, namely, correct, partially correct and erroneous. Reconstructed buildings that have the same shape with their actual counterpart are deemed correct. For the partially correct, they represent the group of connected buildings, where only a portion is successfully reconstructed. The reconstruction is erroneous, when the building model is inherently different in shape with the actual one. Table 1 shows the success rate for the three types of buildings. Seventy six percent of the buildings is correctly reconstructed. The buildings that failed in the reconstruction, i.e., the erroneous category, are the small ones that do not have enough available LIDAR points. The mean value of the height differences between the LIDAR points and roof surface, which is called the shaping error, is 0.12 m. The discrepancies range from 0.06 m to 0.33 m.

5 Conclusions

In this investigation, we have presented a scheme for the extraction and reconstruction of building models via the merging of LIDAR data and aerial imagery. The results from the tests demonstrate the potential of reconstructing the buildings with straight lines and curvilinear boundaries. The building models generated by the proposed method take advantage of the high horizontal accuracy from the aerial image, and high vertical accuracy of the LIDAR data. More than 91% of the building regions are correctly detected by our approach. Among the successfully detected buildings, ninety percent of the buildings are fully or partially reconstructed. The planimetric accuracy of the building boundaries is better than 0.8m, while the shaping error of the reconstructed roofs in height is 0.14 m. This demonstrates that the proposed scheme proves to be a promising tool for future applications.

Acknowledgments. This investigation was partially supported by the National Science Council of Taiwan under Project No. NSC94-2211-E-008-028. The authors would like to thank the Department of Land Administration of Taiwan and Industrial Technology Research Institute of Taiwan for providing the test data sets.

References

1. Hamilton, A., Wang, H., Tanyer, A.M., Arayici, Y., Zhang, X., Song, Y.: Urban Information Model for City Planning. ITcon. 10 (2005) 55-67.
2. Rottensteiner, F.: Automatic Generation of High-quality Building Models from LIDAR Data, IEEE Computer Graphics and Applications. 23 (2003) 42-50.
3. Nakagawa, M., Shibasaki, R., Kagawa, Y.: Fusion Stereo Linear CCD Image and Laser Range Data for Building 3D Urban Model. International Achieves of Photogrammetry and Remote Sensing. 34 (2002) 200-211.
4. Guo, T.: 3D City Modeling using High-Resolution Satellite Image and Airborne Laser Scanning Data. Doctoral dissertation, Department of Civil Engineering, University of Tokyo, Tokyo. 2003.
5. Vosselman, G.: Fusion of Laser Scanning Data, Maps and Aerial Photographs for Building Reconstruction. International Geoscience and Remote Sensing Symposium. 1 (2002) 85-88.
6. Rottensteiner, F., Jansa, J.: Automatic Extraction of Building from LIDAR Data and Aerial Images. International Achieves of Photogrammetry and Remote Sensing. 34 (2002) 295-301.
7. Fujii, K., Arikawa, T.: Urban object reconstruction using airborne laser elevation image and aerial image, IEEE Transaction on Geoscience and remote sensing. 40 (2002) 2234-2240.
8. Rau, J. Y., Chen, L. C.: Robust Reconstruction of Building Models from Three-Dimensional Line Segments. Photogrammetric Engineering and Remote Sensing. 69 (2003) 181-188.
9. Haralick, R.M., Shaunmmugam, K., Distein, I.: Texture Features for Image Classification. IEEE Transactions on Systems Man and Cybernetics. 67 (1973) 786-804.
10. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 8 (1986) 679-698.
11. Hough, P.V.C.: Methods and Means for Recognising Complex Patterns. U.S. patent No. 306954. (1962).
12. Xu, L., Oja, E., Kultanen, P.: A New Curve Detection Method: Randomized Hough Transform (RHT). Pattern Recognition Letters. 11 (1990) 331-338.

The Generation of 3D Tree Models by the Integration of Multi-sensor Data

Liang-Chien Chen, Tee-Ann Teo, and Tsai-Wei Chiang

Center for Space and Remote Sensing Research, National Central University
No. 300, Jhong-Da Road, Jhong-Li City, Tao-Yuan 32001, Taiwan
{lcchen, ann}@csrsr.ncu.edu.tw, 93322084@cc.ncu.edu.tw

Abstract. Three-dimensional tree modeling is an important task in the management of forest ecosystems. The objective of this investigation is to reconstruct 3D tree models using LIDAR data and high resolution images. The proposed scheme comprises of three major steps: (1) data preprocessing, (2) vegetation detection, and (3) tree modeling. The data preprocessing includes spatial registration of the airborne LIDAR and high resolution images, derivation of the above ground surface from LIDAR data, and generation of a spectral index from high resolution images. In the vegetation detection, a region-based segmentation and knowledge-based classification are integrated to detect the tree regions. Afterwards, the watershed segmentation is selected to extract the tree crown and heights. In the last step, we use the tree height, tree crown and terrain information to build up the 3D tree models. The experimental results indicate that the accuracy of the extracted individual tree is better than 80%, while the accuracy of the determined tree heights is about 1m.

Keywords: LIDAR, High Resolution Image, Tree Model.

1 Introduction

Forests are an important and valuable resource to the world. Furthermore, research in relevant environmental issues, such as the ecosystem, biodiversity, wildlife, and so forth, require detailed forest information. In light of this, there is an increasing urgency to obtain a comprehensive understanding of forests. Thus, the 3D tree model plays an important role in forest management.

Remote sensing technology has been applied to forest ecosystem management for many years [1]. Most of the research utilizes the spectral characteristics of optical images to detect the forest, and delineate the independent tree crown [2] [3]. Previous studies used high resolution images to estimate tree locations, canopy density, and biomass [4]. However, optical images are easily influenced by the topography and weather conditions. In addition, as the ability of optical images to penetrate through the forest area is weak, it is unable to directly capture the 3D forest structure.

Nowadays, LIDAR (LIght Detection And Ranging) systems have become a mature tool for the derivation of 3D information. The LIDAR system is an integration of the Laser Scanner, Global Position System (GPS) and Inertial Navigation System (INS). Its high precision in laser ranging and scanning orientation makes the decimeter

L.-W. Chang, W.-N. Lie, R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 34 – 43, 2006.

accuracy for the ground surface possible. The LIDAR technology provides horizontal and vertical information at high spatial resolution and vertical accuracy. Forest attributes like tree heights can be directly retrieved from LIDAR data. Early studies of LIDAR forest measurements used LIDAR to estimate forest vegetation characteristics, such as canopy cover, forest volume, and biomass [5]. However, although the LIDAR data contains abundant 3D shape features, it lacks the spectral information.

The individual tree extraction from LIDAR data is an important topic for 3D tree modeling. The strategy of individual tree delineation can be divided into the following categories: (1) Pixel-based method [6], (2) Region-based method [7], (3) Contour-based method [8], and (4) Empirical method [9]. In the region-based approach, such as the watershed segmentation, it applies mathematical morphology to explore the geometric structure of trees in an image. The advantage of this approach is that the method may selectively preserve the structural information, while also accomplishing the desired tasks within the image. In the region-based approach, homogeneity regarding the shape and color in the neighborhood is examined in a region growing process. The contour-based approach minimizes the internal energy by weighting the parameters. In the empirical method, it collects a large amount of ground truth pertaining to various forest parameters, such as the tree crown width, tree height, and tree age. Afterwards, the relationship among the tree properties is analyzed. A typical result shows that the tree height and tree crown width is revealed to have a linear relationship after a regression analysis is performed. A review of the rapidly growing literature on LIDAR applications emphasizes the need for data fusion in the processing phase of LIDAR data, which may serve as a method in improving the various features extraction task. Therefore, we conducted an integration of the LIDAR data and high resolution image to build up the forest canopy model.

The objective of this investigation was to construct a 3D forest canopy model using LIDAR data and high resolution image. The proposed method was a data fusion scheme with a coarse-to-fine strategy. The proposed scheme comprised of three major steps: (1) data preprocessing, (2) vegetation detection, and (3) tree modeling. The data preprocessing included space registration of the LIDAR and high resolution image, derivation of the above ground surface from LIDAR data, and generation of a spectral index from high resolution image. In the vegetation detection, a region-based segmentation, followed by the knowledge-based classification was integrated to detect tree regions. Subsequently, we performed a tree crown extraction in the vegetation regions. The watershed segmentation and local maximum search were selected to extract the tree crowns. In the last step, we used the tree height, tree crown, and terrain information to build the 3D tree models. The validation datasets included an orchard test site in Taiwan, located in Tai-Chung, and a forest test site in Finland, situated in Espoolahti.

2 Methodology

The proposed scheme encompassed three parts: (1) data preprocessing, (2) vegetation detection, and (3) tree modeling. The preprocessing included a geometric and radiometric processing. A divide-and-conquer strategy was then incorporated to detect the vegetation followed by the tree modeling. Fig.1 shows the flowchart of the proposed method.

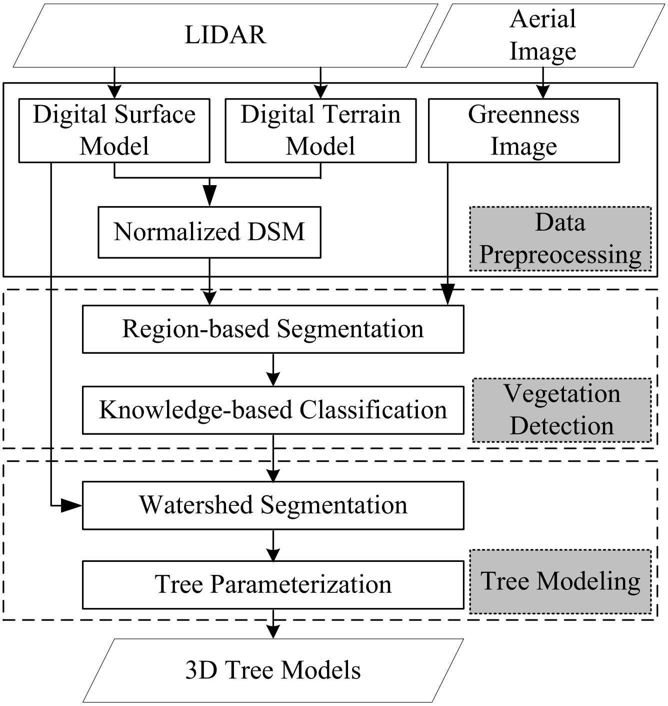


Fig. 1. Flowchart of the proposed method

2.1 Data Preprocsessing

The data preprocessing was composed of three major procedures: (1) space registration, (2) derivation of above ground surface, and (3) generation of a spectral index. The LIDAR data was used to generate a Digital Terrain Model (DTM) and Digital Surface Model (DSM) in grid form. In the space registration, the aerial images were orthorectified using the DSM and ground control points. The orthorectified images were essentially co-registered with the LIDAR data. We subtracted the DTM from the DSM to generate the Normalized DSM (NDSM). The NDSM represented the above ground surface that was used to separate the ground and above ground objects. The formula of NDSM is shown in Eq. (1). Fig. 2 demonstrates a sample result of NDSM. For the multispectral aerial image, we used the red and green bands to calculate the greenness index [10]. Eq. (2) shows the formula of the greenness index. The greenness index was mostly used to identify the vegetation areas. Fig. 3 illustrates a sample result of the greenness image. For cases where the near infrared band was available, the NDVI (Normalized Difference Vegetation Index) was utilized, as it was deemed a better index in accessing the greenness level. The calculation of the NDVI is shown in Eq. (3).

$$NDSM = DSM - DTM \,. \tag{1}$$

$$\text{Greenness} = (G - R) / (G + R) . \tag{2}$$

$$\text{NDVI} = (IR - R) / (IR + R) . \tag{3}$$

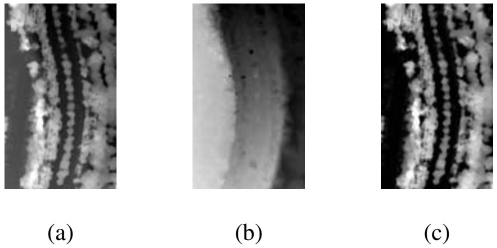


Fig. 2. LIDAR data preprocessing (a) Digital Terrain Model (b) Digital Surface Model (c) Normalized Digital Surface Model

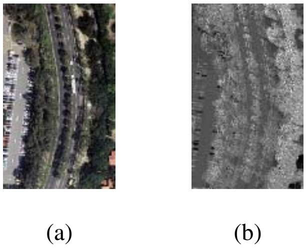


Fig. 3. Aerial image preprocessing (a) aerial image (b) greenness image

2.2 Vegetation Detection

The objective of the vegetation detection was to extract the vegetation areas, in which the non-vegetation areas would not interfere in the tree modeling. We integrated the region-based segmentation and knowledge-based classification during this stage. The elevation from the LIDAR data and radiometric features from the orthoimages were combined in the segmentation. Thus, pixels with similar height and spectral attributes were merged into a region. After the segmentation, each separated region was a candidate object for classification. Considering the height and spectral characteristics, the vegetation areas were extracted by a fuzzy logic classification.

We first used a multiple data segmentation technique to perform the region-based segmentation. It could identify objects with correlated characteristics in terms of reflectance and height. In this step, we fused the NDSM and the greenness level for segmentation. This method identified geographical features using scale homogeneity parameters [11], which were obtained from the spectral reflectance in the RGB and elevation value in the NDSM. The homogeneity was described by a mutually exclusive interaction between the attribute and shape. Eq. (4) shows the formula of the homogeneity index. The composed homogeneity index was based on the attribute and shape factor, where it considered the attribute and shape information simultaneously. The formulas of the attribute and shape factor are shown in Eq. (5) and Eq. (6), respectively. The weights of the attribute and shape factor should be set properly. The attribute factor used the standard deviation of the region as a segmentation criterion. The shape factor selected the smoothness and compactness of the region boundary for weighting. The formulas for the calculations of smoothness and compactness are shown in Eq. (7) and Eq. (8), respectively.

After the segmentation, we employed the object-oriented classification in the vegetation detection. The classification was based on a fuzzy logic classification system, where the membership functions employed thresholds and weights for each data layer. The above ground and high greenness objects were classified as vegetation. Fig. 4 shows an example of the segmentation and classification.

$$H = w * h_{attribute} + (1 - w) * h_{shape} . \qquad (4)$$

where,

H: homogeneity index,
$h_{attribute}$: attribute factor,
h_{shape}: shape factor, and
w: weighting for attribute and shape factor.

$$h_{attribute} = \sum w\_c * \sigma_c . \qquad (5)$$

where,

$h_{attribute:}$ attribute factor,
w_c: weighting among layers, and
σ_c: standard deviation of pixel attribute in a region.

$$h_{shape} = w_s * h_{smooth} + (1 - w_s) * h_{compact} . \qquad (6)$$

$$h_{smooth} = L / B . \qquad (7)$$

$$h_{compact} = L / N^{\mathbf{0.5}} . \qquad (8)$$

where,

$h_{shape;}$ shape factor,
w_s: weighting for smoothness and compactness,
h_{smooth}: smoothness of region,
$h_{compact}$: compactness of region,
L: border length of region,
B: shortest border length of region, and
N: area of region.

2.3 Tree Modeling

After the vegetation detection, we focused on the vegetation areas for the extraction of tree models. The primary process included individual tree segmentation and tree parameterization. We first applied the watershed segmentation method [12] on the DSM in the vegetation regions. To avoid the errors of the DTM, we perform segmentation in the DSM rather than the NDSM. As it could detect the changes of the individual tree height, it was able to extract the boundary of each individual tree. We assumed that the maximum point in the boundary was the tree position. The local maximum method was applied to extract the tree location and tree height. In this study, a least squares circle fitting was applied to extract the tree crown radius. The tree parameters for each individual tree included the tree crown radius, position and height. The 3D tree models were represented by using a tree model database. Once the tree type was selected, the 3D tree model could be generated by the tree parameters [13]. In this study, the tree types are selected manually. Fig. 5 demonstrates a sample procedure of tree modeling.

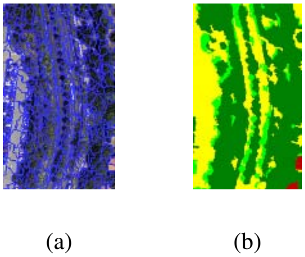


Fig. 4. Vegetation detection (a) segmentation results (b) classified results

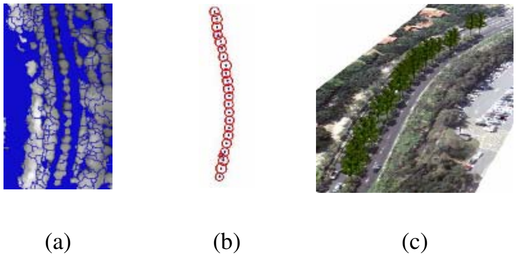


Fig. 5. Tree modeling (a) watershed segmentation results (b) parameterized tree crowns (c) 3D tree models

3 Experiment Results

Two test sites were selected for validation---an orchard area and a forest area. The orchard area was located in Taiwan's Tai-Chung city, and the LIDAR data was obtained via the Optech ALTM2033. The average density of the LIDAR data was roughly 1.7 points per square meter. Color aerial photo with a scale of 1:5,000 were used in test area 1, where it was scanned in a 20 μm per pixel mode. Thus, the ground resolution of the digital images was around 0.1m. The second test area was located in Espoolahti of Finland. The Finland data was released by the European Spatial Data Research (EuroSDR) and International Society for Photogrammetry and Remote Sensing (ISPRS) as a sample test site. The LIDAR data was also obtained by the Optech ALTM2033, and the average density of the LIDAR data was about 2 points per square meter. The Vexcel Ultracam-D image with a 1/6,000 scale was selected for the forest area. The ground resolution of the digital images was about 0.2m. The test data sets are shown in Fig. 6. Since the image included the near infrared band, the NDVI was employed in the Espoolahti case.

In the region-based segmentation, we first set the weights of the image layers on the homogeneity segmentation using eCognition [11]. The weighting of the LIDAR part and aerial photo was 2:1. Considering that the shape of the forest was irregular, the attribute factor was more important than the shape factor. Hence, the attribute and shape factors were 0.8 and 0.2, respectively. After segmentation, we performed the object-oriented classification to determine the tree regions. We used the extracted tree regions in the individual trees extraction. Assuming that the highest point in the boundary was a treetop, one segmented boundary was selected to represent one tree crown. Afterwards, we used the least squares circle fitting to extract the tree crown radius. Finally, the 3D tree models were generated by the tree parameters.

We used stereoscopic measurements to determine the tree heights and tree crowns, as references for validations. The experiments included three different aspects in the validation. The first evaluated the detection rate for an individual tree. The second checked the correctness of the tree crown. The third assessed the accuracy of the tree heights.

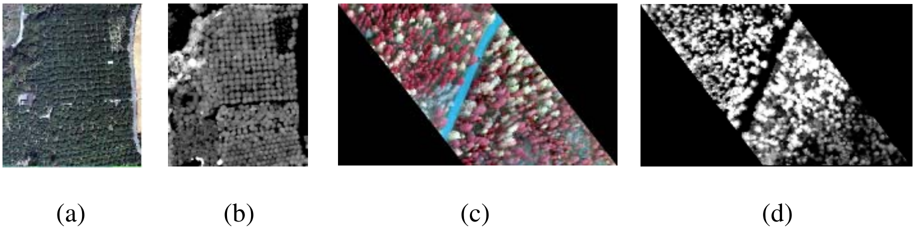


Fig. 6. Test area (a) aerial image for Tai-Chung area (b) NDSM for Tai-Chung area (c) aerial image for Espoolahti area (d) NDSM for Espoolahti area

3.1 Tai-Chung Case

In the Tai-Chung case, we identified objects with correlated characteristics in terms of the reflectance and height. After the vegetation detection, we used tree areas to perform the watershed segmentation on the DSM. Fig. 7a shows the classification results. Fig. 7b reveals the block results, which were derived from the watershed segmentation. The background image is the corresponding DSM. Subsequently, we used the least squares circle fitting to extract the tree crown radius. The results are shown in Fig 7c, and the generated 3D tree models are shown in Fig 7d.

During the evaluation, we checked each segmentation boundary to examine the reliability of the tree crown determination. For the validated patches, we compared the tree heights for accuracy assessments. In this case, we had 197 trees in the reference data. The number of detected trees was 210. The correctness was, thus, 95%. The commission and omission errors stood at 10% and 4%, respectively. Fig. 7c shows the locations of those trees. We then used the 197 matched boundaries to determine the local maximum height for the analysis of the tree crown accuracy. Comparing the automatic results and manual measurement tree crown, the accuracy of the tree crown reached 92%. The commission and omission errors were 29% and 7%, respectively. The RMSE of the tree height was 0.62 m. The result demonstrated a great potential in employing multi-sensor data for 3D tree modeling.

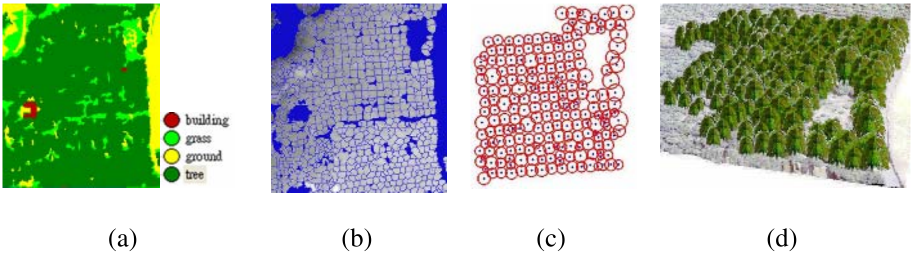
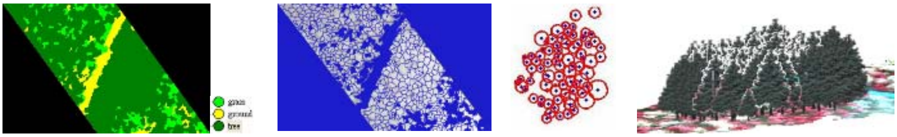


Fig. 7. Results of Tai-Chung area (a) classified results (b) individual tree segmentation results (c) parameterized tree crowns (d) 3D tree models

3.2 Espoolahti Case

The Espoolahti case was a forest area with complex trees. The vegetation results are shown in Fig 8a. The result indicates that most of the areas in the Espoolahti case were tree areas. Fig. 8b is the results of the individual segmentation from DSM. We then selected a small area, as shown in Fig 8b to do the tree crown parameterization. The tree crown is shown in Fig. 8c, and the generated 3D tree models are revealed in Fig 8d.



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

Fig. 8. Results of Espoolahti area (a) classified results (b) individual tree segmentation results (c) parameterized tree crowns (d) 3D tree models

In this case, we had 105 trees in the reference data. The number of detected trees was 97. The number of matched trees was 78, where the correctness was 80%. The commission and omission errors stood at 19% and 25%, respectively. Afterwards, we used the 78 matched boundaries to determine the local maximum height in accessing the accuracy. Comparing the automatic results and manual measurement tree crown, the accuracy of the tree crown reached 78%. The commission and omission errors were 37% and 21%, respectively. The RMSE of the tree height was 1.12 m. As this location was more complicated than the orchard area, the accuracy was lower than the previous case.

Table 1. Summary of tree modeling accuracy

Evaluation Item		Tai-Chung	Espoolahti
Tree detection rate	Accuracy (%)	95.4	80.4
	Commission Error (%)	10.5	25.7
	Omission Error (%)	4.6	19.6
Tree crown evaluation	Accuracy (%)	92.8	78.4
	Commission Error (%)	29.8	37.7
	Omission Error (%)	7.1	21.6
Tree height evaluation	Root-Mean-Squares Error (m)	0.62	1.12
	Error Range (m)	-2.0 ~ 1.8	-1.9 ~ 2.0

3.3 Summary

The accuracy and reliability of our study is shown in Table 1. The experimental results are summarized as follows.

(1) The coarse-to-fine strategy detects the vegetation areas in the first stage. The detected areas are then pinpointed to extract the individual trees. This strategy may reduce the amount of errors involved in the individual trees extraction. The detection rate reaches approximately 80%.

(2) In the individual tree segmentation, we select a morphology filter to extract the tree boundaries from the Digital Surface Model rather than optical image data. The reliability of tree crowns may reach 80%.
(3) Comparing the extracted tree heights and manually measured tree heights. The accuracy of the tree height is about 1m.
(4) The results show the potential of 3D tree model generation by fusing the LIDAR and image data.

4 Conclusions

This investigation presents a scheme of 3D forest modeling by the fusion of spectral and height information. The results indicate that the correctness of the tree extraction reaches 80%, and the accuracy of extracted tree heights is about 1m. Considering the errors in the photogrammetric measurements for the reference data set, the accuracy could be underestimated. The results demonstrate that the proposed scheme may be used to estimate the tree height on individual tree level basis.

Acknowledgments. This investigation was partially supported by the National Science Council of Taiwan under Project No. NSC94-2211-E-008-027. The authors would like to thank the Council of Agriculture of Taiwan, European Spatial Data Research (EuroSDR) and International Society for Photogrammetry and Remote Sensing (ISPRS) for providing the test data sets. The authors also wish to thank the anonymous reviewer for helpful comments and suggestions.

References

1. Lefsky, M.A., Cohen, W.B., Spies, T.A.: An Evaluation of Alternate Remote Sensing Products for Forest Inventory, Monitoring, and Mapping of Douglas-Fir Forests in Western Oregon. Canadian Journal of Forest Research. 31 (2001) 78–87.
2. Gong, P., Mei, X., Biging, G.S., Zhang, Z.: Improvement of an Oak Canopy Model Extracted from Digital Photogrammetry. Photogrammetric Engineering and Remote Sensing. 63 (2002) 919-924.
3. Wang, L., Gong, P., Biging, G.S.: Individual Tree-Crown Delineation and Treetop Detection in High-Spatial-Resolution Aerial Imagery. Photogrammetric Engineering and Remote Sensing. 70 (2004) 351-357.
4. Sheng, Y., Gong, P., Biging, G.S.: Model-Based Conifer Canopy Surface Reconstruction from Photographic Imagery: Overcoming the Occlusion, Foreshortening, and Edge Effects. Photogrammetric Engineering and Remote Sensing. 69 (2003) 249-258.
5. Popescu, S.C., Wynee, R.H., Nelson, R.F.: Measuring Individual Tree Crown Diameter with Lidar and Assessing Its Influence on Estimating Forest Volume and Biomass. Canadian Journal of Remote Sensing. 29 (2003) 564–577.
6. Yu, X., Hyyppä, J., Kaartinen, H., Maltamo, M.: Automatic Detection of Harvested Tees and Determination of Forest Growth using Airborne Laser Scanning. Remote Sensing of Environment. 90 (2004) 451-462.

7. Suárez, J.C., Ontiveros, C., Smith, S., Snape, S.: Use Of Airborne Lidar and Aerial Photography in the Estimation of Individual Tree Height in Forestry. Computer and Geosciences, 31 (2005) 253-262.
8. Persson, A., Holmgren, J., Soderman, U.: Detecting and Measuring Individual Trees using an Airborne Laser Scanner. Photogrametry Engineering and Remote Sensing. 68 (2002) 925-932.
9. Popescu, S.C., Wynne, R..H.: Seeing the Trees in the Forest: Using LIDAR and Multispectral Data Fusion with Local Filtering and Variable Window Size for Estimating Tree Height. Photogrametry Engineering and Remote Sensing. 70 (2004) 589-604.
10. Niederöst, M.: Automated Update of Building Information in Maps using Medium-Scale Imagery (1:15,000). In: Baltsavias, E., Gruen, A., van Gool. L. (eds.): Automatic Extraction of Man-Made Objects from Aerial and Space Images. 3 (2001) 161-170.
11. Definiens: eCoginition User's Guide. (2004) 79-81.
12. Luc V., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence. 13 (1991) 583-598.
13. Dunbar M.D., Moskal, L.M., Jakubauskas, M.E.: 3D Visualization for the Analysis of Forest Cover Change. Geocarto International. 19 (2004) 103-112.

LOD Generation for 3D Polyhedral Building Model

Jiann-Yeou Rau[1], Liang-Chien Chen[2], Fuan Tsai[3],
Kuo-Hsin Hsiao[4], and Wei-Chen Hsu[4]

[1]Specialist, [2] Professor, [3]Assistant Professor,
Center for Space and Remote Sensing Research,
National Central University, Jhong-Li, Taiwan
{jyrau, lcchen, ftsai}@csrsr.ncu.edu.tw
[4] Researcher, Energy and Environment Lab.,
Industrial Technology Research Institute, Chu-Tung, Taiwan
{HKS, ianhsu}@itri.org.tw

Abstract. This paper proposes an algorithm for the automatic generation of Levels-of-Detail (LODs) for 3D polyhedral building models. In this study a group of connected polyhedrons is considered as "one building", after which the generalization is applied to each building consecutively. The most detailed building models used is the polyhedral building model which allows for an elaborate roof structure, vertical walls and a polygonal ground plan. In the work the term "Pseudo-Continuous LODs" is described. The maximum distinguishable "feature resolution" can be estimated from the viewer distance to a building and is used to simplify the building structure by the polyhedron merging and wall collapsing with regularization processes. Experimental results demonstrate that the number of triangles can be reduced as a function of the feature resolution logarithm. Some case studies will be presented to illustrate the capability and feasibility of the proposed method including both regular and irregular shape of buildings.

Keywords: Levels-of-Detail, Generalization, 3D Building Model.

1 Introduction

The creation of a digital 3D city model is a generalization procedure from a complex world to digital geometric data that can be stored in a computer. Important spatial features such as roads, buildings, bridges, rivers, lakes, trees, etc. are digitized as two or three dimensional objects. Among them, the most important one is the 3D building models. From the application point of view, the more detail provided by a building model the greater the amount of geometrical information available for spatial analysis and realistic visualization. However, since the computer has limited resources for computation, storage and memory, the geometry of building models has to be generalized to reduce their complexity and increase their efficiency of demonstration or analysis. This means that th.e generation of different LODs of 3D building models for real-time visualization applications must be a compromise between structural detail and browser efficiency.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 44–53, 2006.

In computer graphics, the efficiency of 3D browsing is highly dependent on the number of triangles and textures to be rendered. Many algorithms regarding terrain simplification have already been developed in the field of computer vision and computer graphics. For example, Hoppe [1] and Garland & Heckbert [2] introduced an edge collapse transformation to simplify surface geometry, resulting in continuous LODs that can be applied for progressive meshing applications. In the case of 3D building models generalization, Sester [3] proposed a least squares adjustment method to simplify building ground plans. Kada [4] adopted a similar concept and applied it to 3D polyhedral building models. Based on the scale-spaces theory, Mayer [5] has suggested using a sequence of morphological operations to generalize 3D building models. However, the method is suitable for orthogonal building models only. In case of non-rectangular buildings they have to be squared in advance before generalization.

In this paper, the concept of "Pseudo-Continuous LODs" (PCLODs) is utilized for the generalization of 3D polyhedral building models. The generation of Continuous LODs for a 3D city model is impractical for the following two reasons. The first reason is that the buildings are mostly separated by some distance so their geometry is different from a digital terrain model. The second reason is that buildings have vertical walls and the topology has to be maintained during geometric simplification. Thus in this work we treat a group of connected or contacted polyhedrons as "one building model" and the generalization is applied to each building model respectively.

The idea for PCLODs comes from the digital camera. The computer screen corresponds to the camera's focal plane with a fixed CCD sensor size, frame area and focal length. In computer rendering and simulation of a virtual landscape is similar to the photo imaging process. A building's structural details make it easy to recognize when the viewer distance is closer to the target, which is similar to rendering on the computer screen. On the contrary, when the viewer distance to the target gets longer, the smaller structure of buildings become indistinguishable they can thus be ignored during computer rendering.

Here we focus on the generalization of 3D polyhedral building models. The most detailed "3D polyhedral building model" is generated semi-automatically based on the SPLIT-MERGE-SHAPE (SMS) algorithm [6]. The SMS algorithm utilizes 3D building structure lines that have been measured from aerial stereo-pairs. The 3D structure lines can be either derived from semi-automatic stereo-measurements [8] or extracted from high-resolution satellite images and/or aerial imagery [9]. Buildings with gable roofs, flat roofs, and regular or irregular ground plan can be described. Two neighboring buildings will not overlap due to the SPLIT process provided by the SMS algorithm.

In a 3D visualization system, the viewer distance can be used to estimate the maximum distinguishable "feature resolution". The feature resolution is used to detect small 3D features in one building model. Since the distance variation in dynamic 3D browsing is continuous, the corresponding PCLOD 3D building models can thus be generated immediately. However, in order to reduce the computational load during generalization, we can adopt out-of-core processing, by generating all the LODs of a building model before 3D visualization. Some case studies will be presented to illustrate the capability and feasibility of the proposed method for both regular and irregular type of buildings.

2 Methodology

The proposed scheme for generating PCLODs for a 3D building model can be divided into two parts, i.e. polyhedron merging and wall collapsing with regularization. Fig. 1 illustrates a flowchart of the proposed approach. In which we iteratively simplify the original detailed building model or generalized building model by enlarging the values of the feature resolution. The result is a sequence of different LODs of 3D building model from fine to coarser. The method can be used for out-of-core processing or real-time visualization, depending on the power of the computer, the number of building models to be generalized, and the required rendering frame rate. In the following sections, the definition of feature resolution and the two major procedures are described.

2.1 Feature Resolution

In the proposed generalization, only one parameter is adopted for geometry simplification, i.e. the feature resolution (R). During 3D browsing, we can calculate the distance between the viewer and the building. The center of a group of polyhedrons is used in the distance calculation to maintain consistency within a building. According to the perspective projection geometry, we can simulate the computer screen as a CCD frame or a digital camera. Using the viewer distance and the camera's focal length, we can estimate the scale between the object space and the image plane. The feature resolution can thus be obtained using the scale and the CCD cell size. It is also similar to a ground sampling distance (GSD) of one pixel during the photo imaging process.

In 3D computer graphics, ground features with a size smaller than the feature resolution are indistinguishable and may be eliminated in order to simplify the geometry and increase the rendering efficiency. In this study, the feature resolution is given and insignificant geometrical structures are detected and eliminated.

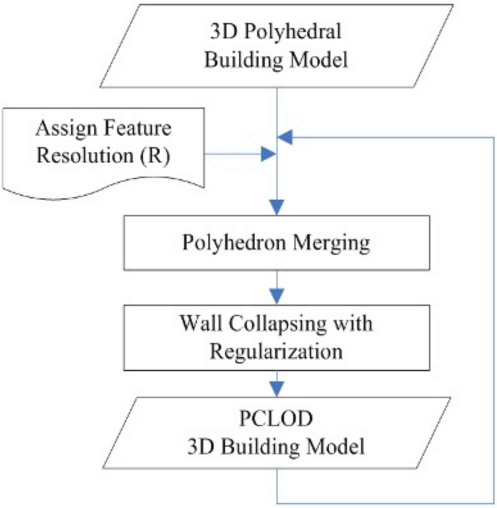


Fig. 1. The proposed generalization scheme

2.2 Polyhedron Merging

The polyhedron merging process includes the following three procedures, i.e. (1) the flattening of sloped roofs, (2) the merging of two connected polyhedrons with a small height difference, and (3) the elimination at small polyhedrons contained within a bigger polyhedron. The first two steps are performed only on the building's roof structure. Two visual important features are considered, i.e. the length of the sloped roof-edge and the height difference between two neighboring flat-roofs.

The flattening of sloped roofs is processed before the merging of two neighboring flat-roofs. There are many kinds of roof types, gable, hipped, tent-shaped, hexagonal, pyramid-shaped roofs and so on, that have to be considered as visual important features. In the polyhedral building model, these roof structures are constructed by sloped triangles with height variations which are different from those at horizontal rooftops. In cases when the length of the edge of a roof-triangle is smaller than the feature resolution, the sloped roof-triangle will be flattened. This means that their roof height is replaced by the average of the original ones. For example, Fig.2 (A) shows a building with two gable roofs and two flat roofs. Utilizing a feature resolution of 3 meters, these two gable roofs will be flattened at first, as shown in Fig.2 (B).

The second step is to merge two neighboring flat-roofs with an insignificant height difference. If their height difference is smaller than the feature resolution, the neighboring polyhedrons will be merged as one. As illustrated in Fig.2 (C), there are five polyhedrons that are merged into one polyhedron. The final roof-height will be close to the roof with the larger area, as obtained by the weighting average of two roof-heights via the horizontal projection area.

In order to compare the height difference between two polyhedrons, the topology of all the polyhedrons has to be constructed first. The topology of a polyhedron provides information regarding its neighboring polyhedrons and its corresponding walls. The walls can be categorized as either *independent walls* or *shared walls*. A shared wall means there is a corresponding neighboring wall, but this is not true for independent walls.

In an island type of building, one small polyhedron will be contained within a bigger polyhedron. The third step is thus designed to eliminate the interior small polyhedron. If the length of all the walls of the interior polyhedron and the height difference between these two polyhedrons are all smaller than the feature resolution, the interior one will be directly removed.

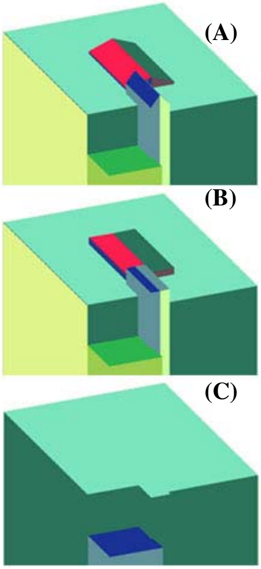


Fig. 2. Example of polyhedron merging

2.3 Wall Collapsing with Regularization

Since most of buildings are composed of at least four walls, i.e. a tetrahedron, a building with less than six walls is not considered in the proposed generalization scheme. In order to preserve the principal structure of a building during generalization, the longest walls have to be detected and used for shape regularization. Using the independent walls as described in the previous section, we can construct the building's boundaries. The principal structure analysis is applied on the boundary only. Adopting the concept of visual importance, the longest walls are considered as the principal structure of a building. Thus, all independent walls are fist sorted according to length so that a threshold to separate them from major and minor structures can thus be determined. The threshold is determined by the average length of the fourth and the fifth longest walls. Any wall with a horizontal length longer than the threshold is assigned as the principal structure of the building. In the simplification process, the principal structure will be maintained, in order to maintain the visual importance of a building between two consecutive LODs.

The purpose of the above polyhedron merging process is mainly to generalize the rooftop structure. The result maintaining a detailed building ground plan, including intrusions or pillars located at building corners. In 1996, Hoppe [1] introduced an edge collapse transformation method to eliminate two neighboring triangles resulting in the merging of two vertices into one. This operation cannot be directly applied to simplify a 3D polyhedral building model. For example, Fig.3 (A) illustrates three connected vertical walls where each wall is composed of two triangles. If the roof-top edge on the middle wall is contracted by merging two corners into one, the wall becomes sloped and destroys the topology as a vertical wall, as shown in Fig.3 (B). In this paper, we adopt the same concept by contracting the whole wall not only one edge. The effect is illustrated in Fig.3 (C)-(D) using the same example. The purpose of wall collapsing is to simplify the geometric building structure by detecting and eliminating insignificant features.

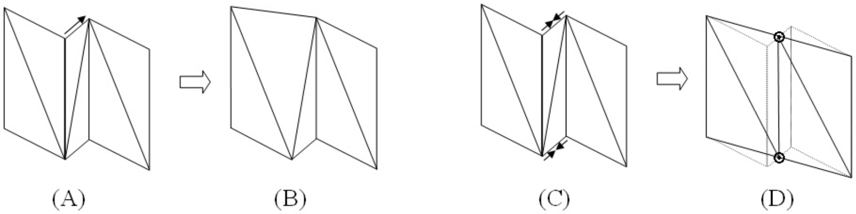


Fig. 3. Wall collapse operation for 3D building model

Before the wall collapsing procedure, we need to detect insignificant geometric structures by searching for all short walls in one polyhedron. The shortest wall is eliminated one by one up to a certain threshold length, which is the same as in principal structure analysis. However, in order to retain main structure of the building, additional co-linear processing is necessary. During wall collapsing independent walls and shared walls are treated separately.

- For a sequence of shared walls neighboring the same polyhedron has to be processed at the same time. A fixed point is first determined at the center of an insignificant wall, as shown in Fig.3 (D) denoted by the symbol "⊙". The endpoint of the previous wall and the starting-point of the following wall are merged at the fixed point. In cases where the previous wall or the following wall is related to different polyhedron, the fixed point is determined by the junction point of the two walls which correspond to two neighboring polyhedrons. This will avoid a gap between them when a center location is used as a fixed point.

For example, Fig. 4 illustrates three polyhedrons projected on the horizontal plane. In Fig. 4(A), the shared walls are denoted by red lines, in which E1 is the detected insignificant wall to be eliminated. In Fig. 4(B), if we directly extract E1 by its center, i.e. T1, a gap will be occurred between the left polyhedron and the right two polyhedrons. So, during wall collapsing we choose the junction point T2, as indicated in Fig. 4(C), as the fixed point to avoid the gap effect.

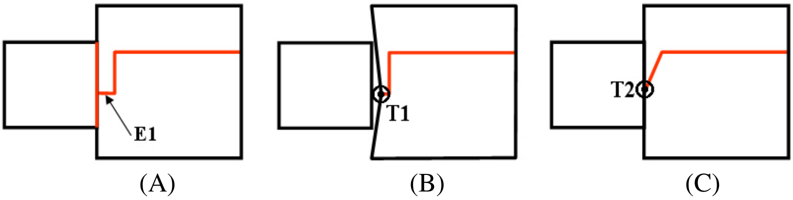


Fig. 4. Prevention of gap effect during wall collapsing

- For independent walls, we have to maintain the principal structure of a building. Thus, the vertices of a wall that belong to the principal structure cannot be moved during the wall collapsing process. This means that the fixed point is set at the junction of the insignificant wall and the principal wall. Since independent walls construct the building boundary, the visual effect will be better and the number of walls can be reduced further by applying co-linear processing along the principal structure. A piping technique is utilized to accomplish this procedure. A wall that belongs to the principal structure is chosen as the pipe axis. The radius of the pipe is the same as the feature resolution. Any wall that was contained in the pipe is projected onto the pipe axis, except for wall vertices that are connected to a shared wall, again to avoid the gap effect.

For example, Fig.5 illustrates two polyhedrons projected on the horizontal plane. The red lines denote the principal structure. In Fig. 5(A), E1~E3 are three detected insignificant walls to be eliminate. Since T1 is a termination point related to the principal structure the movement is not allowed, during collapsing of E1 the fixed point is set at T1. The elimination of E2 has the same situation as eliminating E1 that T2 is set as a fixed point. However, for the extraction of E3, the fixed point is set at the center of E3, i.e. T3. Fig. 5(B) demonstrates the wall collapsing result. In the following regularization process, we choose E0 as the pipe's axis. The pipe is denoted by two green dashed lines, as shown in Fig. 5(B). Although E4 and E5 are all contained in the pipe, E4 is connected to a shared wall the movement is not allowed. On the other hand, E5 has to be projected onto the pipe's axis to finish the co-linear process, as show in Fig. 5(B-C) T3 has moved to T5.

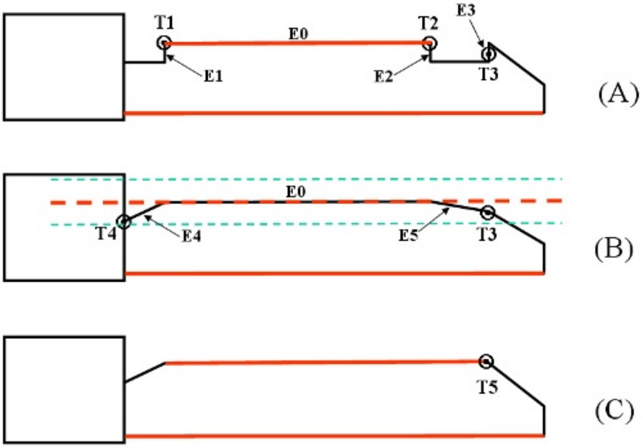


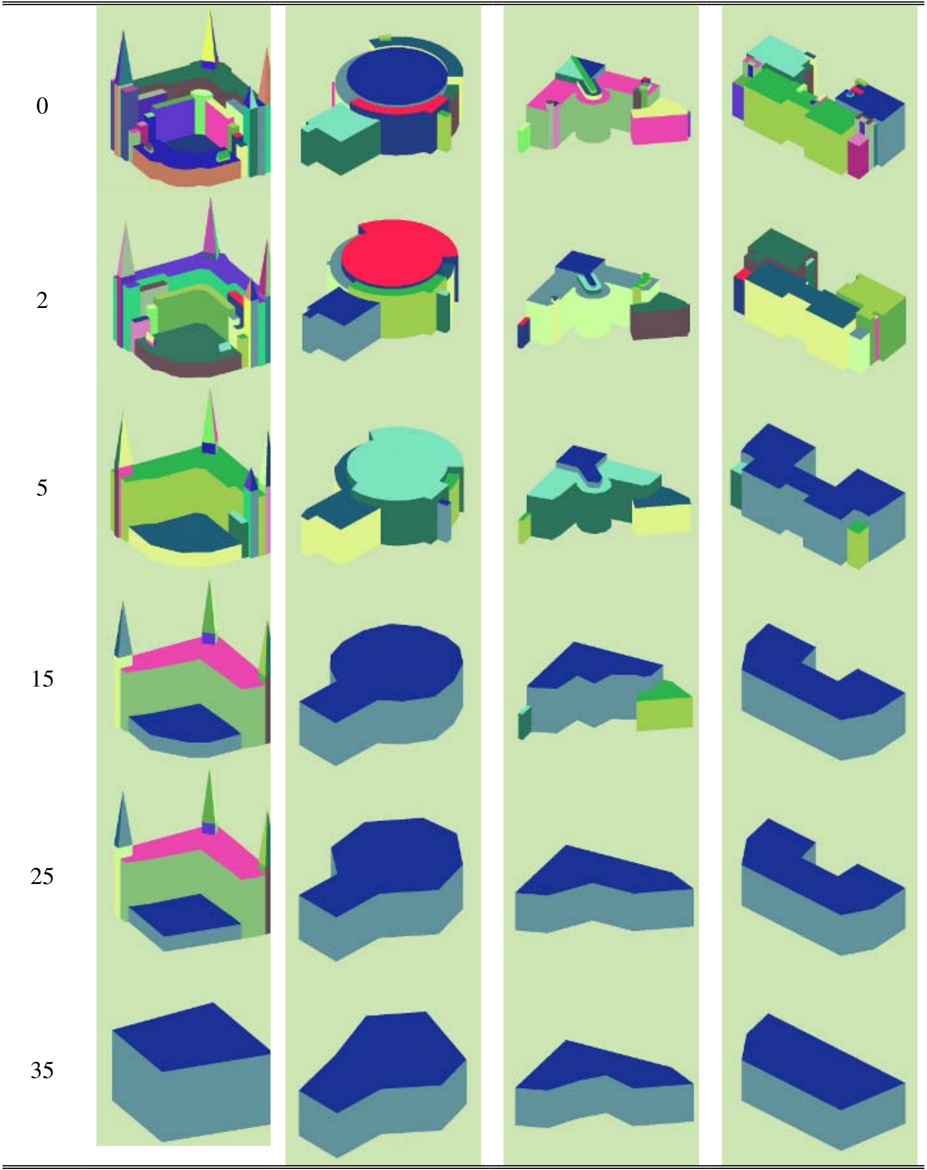
Fig. 5. Wall collapsing with regularization

3 Case Study

The total number of triangles is an important index for evaluating the efficiency of computer graphics rendering. This is especially crucial when the viewer distance becomes longer that the number of buildings within the view frustum will be increased significantly. In this section we like to estimate the reduction rate for when the feature resolution (R) is getting larger, which corresponds to the viewer moving farther away from the target.

In the experiment, the total number of triangles is the sum of all roofs [9] and walls triangles. Four complex building models are utilized as examples, as shown in Fig. 6. The feature resolution is changed from one meter to 35 meters, in which only six of 50 LODs are illustrated for demonstration. In the figure, the original elaborate 3D polyhedral building model is denoted by R equal to zero, while the other LODs are generated using a feature resolution of 2, 5, 15, 25, and 35 meters, respectively. The results demonstrate that the principal structure of the building is preserved for all LODs. In addition, two irregular shaped buildings are test in the experiments, i.e. case 2 and case 3, the feasibility of the proposed scheme is also demonstrated.

Since the generalization result is dependent on the complexity of the building models two consecutive LODs may have the same geometry, i.e. same number of triangles. For example, the generalization results are exactly the same when using R equals to 15 and 25 in Case 4. In Fig.6, the maximum and minimum number of total triangles is also illustrated. From which, one may notice that the ratio of the maximum and the minimum are from 27 to 53. This means that the total number of triangles is reduced by more than 27 times compares to the original one. The implication is that during 3D visualization of a city model, more than 27 times the number of building models could be rendered in real-time when the viewer is at long distance away.



R	Case 1	Case 2	Case 3	Case 4
Max.	535	721	731	434
Min.	10	16	19	16

Fig. 6. Four case studies of generated PCLODs of 3D building models, where R is the feature resolution in meters, and Max./Min. indicate the maximum and minimum number of triangles for all PCLODs of 3D building models

Fig.7 illustrates the total number of triangles (y-axis) vs. the feature resolution (x-axis). A logarithmic function (Ln) is chosen to fit the relation between the feature resolution and the total number of triangles in the building model. The fitting is applied only in the range between the maximum and minimum total of triangles, for each case respectively. The results are indicated in Fig.7 with a dashed line. The correlation coefficients after regression for all cases are above 0.8. This demonstrates that the triangle reduction rate is high especially at smaller feature resolution.

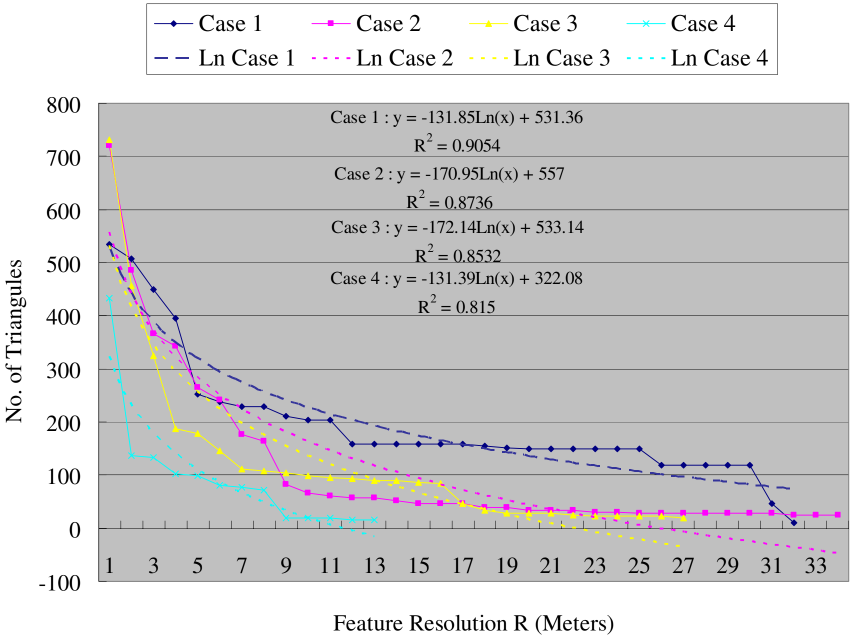


Fig. 7. Total number of triangles vs. feature resolution R for each case. The logarithmic function and its fitting results are indicated by a dashed line with a corresponding correlation coefficient (R^2).

4 Conclusions

This paper presents an automatic generalization approach for 3D polyhedral building models where only one parameter, i.e. the feature resolution, is used. Since a continuous generalization is not applicable to 3D building models, we propose to generate Pseudo-Continuous LODs by changing the feature resolution. The feature resolution can be estimated from the image scale and the virtual CCD size. The number of triangles for a complex building can be significantly reduced as a function of the feature resolution logarithm. As well, the principal structure of the building can be preserved, avoiding the popping effect produced when the LOD is changed. The proposed algorithm can be applied for the generalization of irregular shaped buildings. The experimental results demonstrate that the proposed algorithm is effective in terms of reducing the number of triangles needed and also maintaining the principal structure of a building. A greater than 27 time reduction rate can be achieved for complex building models. The result is applicable for 3D real-time visualization applications of a digital city model. The proposed algorithm can also be used for out-of-core processing or real-time visualization depending on the power of the computer and the number of building models to be generalized. However, the method is not designed for view-dependent generalization or the aggregation of two non-connected buildings. The combination of aggregation in the generalization for adjacent buildings is necessary for future research to further reduce the amount of geometric data. As well, the automatic generation of multi-resolution facade texture from the corresponding building models LOD is necessary for photo-realistic visualization applications.

Acknowledgment

This paper is partially supported by the National Science Council under project numbers NSC 89-2211-E-008-086, NSC 93-2211-E-008-026 and the NCU-ITRI Union Research and Developing Center under project number NCU-ITRI950304.

References

1. Hoope, H., 1996,"Progressive meshes", Proceedings of ACM SIGGRAPH 96, pp.325-334.
2. Garland, M., Heckbert, P., 1997,"Surface simplification using quadric error metrics". In: Proceedings of ACM SIGGRAPH 97, pp.206-216, 1997.
3. Sester, M., 2000, "Generalization based on least squares adjustment", International Archives of Photogrammetry and Remote Sensing, Amsterdam, Netherlands, Vol. XXXIII, Part B4, pp.931-938.
4. Kada, M., 2002,"Automatic generalisation of 3D building models". International Archives of the Photogrammetry, Remote Sensing and Spatial Information Services, Volume XXXIV, Part 4, pp.243-248.
5. Mayer, H., 2005,"Scale-spaces for generalization of 3D building", International Journal of Geographical Information Science, Vol.19, No.8-9, Sep.-Oct. 2005, pp.975-997.
6. Rau, J. Y., and Chen, L. C., 2003, "Robust reconstruction of building models from three-dimensional line segments", Photogrammetric Engineering and Remote Sensing, Vol. 69.No.2, pp. 181-188.
7. Rau, J. Y., Chen, L. C., and Wang, G. H., 2004, "An Interactive Scheme for Building Modeling Using the Split-Merge-Shape Algorithm", International Archives of Photogrammetry and Remote Sensing, Vol. 35, No. B3, pp. 584-589.
8. Chen, L. C., Teo, T. A., Shao, Y. C., Lai, Y. C., and Rau, J. Y., 2004, "Fusion of LIDAR Data and Optical Imagery for Building Modeling", International Archives of Photogrammetry and Remote Sensing, Vol. 35, No. B4, pp. 732-737.
9. Ratcliff, J. W., 2006,"Efficient Polygon Triangulation", Available: http://www.flipcode. com/files/code/triangulate.cpp.

Octree Subdivision Using Coplanar Criterion for Hierarchical Point Simplification

Pai-Feng Lee[1], Chien-Hsing Chiang[1], Juin-Ling Tseng[2],
Bin-Shyan Jong[3], and Tsong-Wuu Lin[4]

[1] Dept. of Electronic Engineering, Chung Yuan Christian University,
[2] Dept. of Management Information System, Chin Min Institute of Technology,
[3] Dept. of Information & Computer Engineering, Chung Yuan Christian University,
[4] Dept. of Computer & Information Science, Soochow University
{allen, hikki, arthur}@cg.ice.cycu.edu.tw,
bsjong@ice.cycu.edu.tw, twlin@cis.scu.edu.tw

Abstract. This study presents a novel rapid and effective point simplification algorithm based on point clouds without using either normal or connectivity information. Sampled points are clustered based on shape variations by octree data structure, an inner point distribution of a cluster, to judge whether these points correlate with the coplanar characteristics. Accordingly, the relevant point from each coplanar cluster is chosen. The relevant points are reconstructed to a triangular mesh and the error rate remains within a certain tolerance level, and significantly reducing number of calculations needed for reconstruction. The hierarchical triangular mesh based on the octree data structure is presented. This study presents hierarchical simplification and hierarchical rendering for the reconstructed model to suit user demand, and produce a uniform or feature-sensitive simplified model that facilitates rapid further mesh-based applications, especially the level of detail.

1 Introduction

Due to the continuing development of computer graphics technology, diversified virtual reality applications are being increasingly adopted. Exactly how three dimensional (3D) objects in the real world can be efficiently and vividly portrayed in virtual scenes has recently become a crucial issue in computer graphics. A triangular mesh is one of the most popular data structures for representing 3D models in applications. Numerous methods currently exist for constructing objects using surface reconstruction, and reconstructing a point cloud as a triangular mesh. The data for sampled points are generally obtained from a laser scanner. However, the extracted sampled points are frequently affected by shape variation, causing over-sampling in the flat surface (Fig. 1). The number of triangles created rises as the number of points sampled from the surface of a 3D object rises, helping to reconstruct the correct model. However, subsequent graphics applications, such as morphing, animation, level of detail and compression, increase the computation costs. Appropriate relevant points should be chosen to retain object features and reduce storage and calculation costs.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 54 – 63, 2006.

Hence, reducing the number of triangles while retaining surface characteristics within a certain error value is worthy of research. The main research on object simplification can be divided into two fields, mesh-based simplification [1, 2, 3, 7, 9, 14] and point-based simplification [5, 6, 8, 10, 12, 13]. Mesh-based simplification requires connectivity relations to be obtained in advance. Hence, the sampled point data acquired by a scanner must perform triangulation by a reconstruction algorithm before simplification. Surface reconstruction extracts sampled points from a 3D object, and reconstructs the triangular mesh from an original object within a certain tolerance level [4]. The mesh-based simplification then attempts to reduce the number of triangles in triangular mesh while maintaining the object quality. Numerous good mesh-based simplification algorithms have been presented, including vertex decimation [14], edge contraction [1], triangle contraction [2], vertex clustering [7], vertex pair contraction [9] and feature extraction [3]. Traditional methods such as Quadric Error Matrices (QEM) [9] have decimation operations that are generally arranged in a priority queue according to an error matrix that quantifies errors caused by decimation. Simplification is performed iteratively to reduce any smoothing of point pairs caused by the decimation operation. This greedy technique can obtain the simplified model with the minimum error of the original model. However, these algorithms all achieve good simplification effect in application, but need a triangular mesh and connectivity in advance of simplification. Restated, the algorithms are burdened with a large number of computations before simplification processing. Consequently, this process is prohibitively expensive.

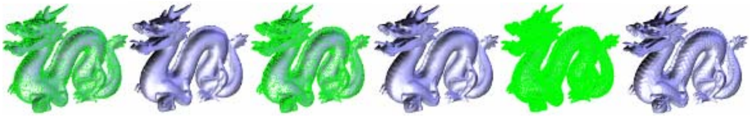


Fig. 1. Over-sampling in the flat region needlessly increases the number of calculations. The simplified model produces the same effect of solid representation.

Therefore, point cloud simplification is an attractive approach. Point-based simplification is applied before reconstruction. If suitable relevant points can be extracted from a point cloud that represent surface variation, then the number of calculations needed for reconstruction can be significantly reduced. Dey [13] presented the first point cloud simplification approach. Dey adopted local feature sizes to detect redundancy in the input point cloud and ensure relevant point densities, thereby exploiting a 3D Voronoi diagram for a densely distributed original point set in advance of simplification. However, this method also requires many computations. Boissonnat and Cazals [6] presented a coarse-to-fine point simplification algorithm that randomly calculates a point subset and builds a 3D Delaunay triangulation. Additional points are inserted iteratively based on their distance to the closest 3D facet until the simplified surface mesh conforms to the restricted error value. Allegre [12] presented a convex hull for all points that adopts a decimation scheme to merge adjacent points according to geometrical and topological constraints. These algorithms must adopt pre-processing to retain the original surface data before simplifying the point set, and therefore require many computations.

Pauly [10] applied the four mesh-based simplification techniques to point cloud simplification. A uniform incremental clustering method is computationally efficient, but leads to a high mean error. The hierarchical clustering method can reduce calculation and memory, but has a marginally better mean error value than the uniform incremental clustering method. The quadric error-based iterative simplification method obtains the best error rate, but has a major disadvantage in that its execution time is sensitive to the input point set size. The particle simulation method obtains a good error rate, but requires many calculations. Alexa [8] proposed to uniformly simplify the point set by estimating the distance from a point to the Moving Least Square (MLS) surface. Alexa also presented a re-sampling method to ensure the distribution of density. Moenning and Dodgson [5] presented an intrinsic coarse-to-fine point simplification algorithm that guarantees uniform or feature-sensitive distribution. They adopted the farthest point sampling and a fast marching algorithm to choose relevant points and set density threshold to ensure point set density. However, their method requires expanding the computational 3D Voronoi diagram, and consequently requires many computations and a large memory.

This study presents a novel rapid and effective point simplification algorithm based on a point cloud without normal and connectivity information. This study initiates with a scattered sampled point set in 3D, and the final output is a triangular mesh model simplified according to restrictive criteria. The proposed method reduces the number of calculations between triangulation and establishing the connectivity relation, and includes three main steps, point simplification, reconstruction guarantee and hierarchical simplification. In the point simplification step, sample points are obtained using 3D acquisition devices that fully represent the object surface variation. This step investigates how to best choose the most appropriate number of relevant points from the sampled points to reduce the complexity of operation and obtain an acceptable simplified result. Sampled points are clustered, based on shape variation, by using the octree data structure, which is an inner point distribution of a cube, to judge whether these points correlate with the coplanar properties. The local coplanar method causes the simplified model to have a feature-sensitive property. The feature-sensitive distribution can achieve a small simplification-based error rate, but does not permit successful reconstruction. Consequently, considering the scattered relevant points in the levels of the hierarchical tree are dynamically adjusted. Moreover, the distribution density is increased by dummy vertices in the region with excessive difference between adjacent levels, helped by hierarchical tree information. The problem of undersampling is thus successfully solved, obtaining a good simplified, reconstructed model. Finally, a hierarchical triangular mesh suitable for multi-resolution is obtained after successfully reconstructing the simplified point set.

This study presents a novel method for extracting the relevant points for a dense input point set, and adopts the reconstruction algorithm presented by Jong [4] to generate a simplified model. Experimental results confirm that a good simplified model, with the advantages given below, can be quickly obtained.

1. Connectivity relations do not need to be recorded. Hence, the reconstruction algorithm can be adopted, significantly reducing the computational cost.
2. The calculations for extracting the relevant points ensure that the simplified model has a good error rate.
3. Using an octree data structure maintains multi-resolution.

4. The hierarchical rendering and imaging can be interactively changed by automatically assigning local sampling constraints.

2 Algorithm Overview

The following steps are crucial to simplifying the sampled point cloud and completely reconstructing the triangular mesh.

2.1 Choose the Coplanar Variable

In this study, the point cloud is subdivided iteratively according to the space coordinates until each cluster meets its respective restricted criterion. The local neighbor of a point set for a cluster was identified by the formula presented by Pauly [10], based on the following equations:

$$c_i \overset{def}{=} \frac{1}{|N_i|} \cdot \sum_{q \in N_i} q \qquad , \; c_i \in R^3 \tag{1}$$

$$C_i \overset{def}{=} \frac{1}{|N_i|} \cdot \sum_{q \in N_i} (q - c_i)(q - c_i)^t \qquad , \; C_i \in R^{3*3} \tag{2}$$

$$f = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \qquad , \; if \qquad \lambda_0 \le \lambda_1 \le \lambda_2 \tag{3}$$

where, given a cluster, c_i denotes the gravity center; C_i denotes the covariance matrix; q denotes the point in a cluster, and $/N_i/$ denotes the number of points.

According to Eq. (2), the covariance matrix C_i is a symmetrical positive semi-definite 3×3 matrix with three eigenvalues λ_0, λ_1 and λ_2. These eigenvalues measure the variation of points in the direction of their corresponding eigenvectors e_0, e_1, e_2. Eigenvector e_0 is a vector characteristic of the minimum eigenvalue λ_0, which denotes the normal vector of a cluster. A cluster conforms to coplanar characteristics if $\lambda_0 << \lambda_1$ and $\lambda_0 << \lambda_2$. Equation (3) determines whether a cluster is subdivided according to the coplanar variable f.

The subdivision criterion for the octree is based on the coplanar variable f, which determines whether a node must be subdivided. This step ensures that dynamic subdivision is performed according to model surface variations. The rough part of model is subdivided, and then additional relevant data can be obtained to refine regions including the object feature regions. For the flat area, a large number of sampled points are reduced to a single point (Fig. 2).

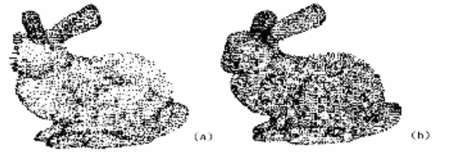


Fig. 2. The Bunny model (*34838* points) as an example using different thresholds. When the coefficient $f = 0.005$ (a), the number of relevant points is *5616*; when $f = 0.001$, the number of relevant points is *12307* (b).

2.2 Choose the Relevant Point of Each Cluster

The point set is subdivided into clusters According to the coplanar variable *f*, and the relevant point from each cluster is then selected. This point denotes the local surface characteristic for the entire cluster. If the selected point is the gravitational center of a cluster, then either it is not the original point, or a major error has occurred, in which case some unexpected triangles may protrude from and concave into the surface (Fig. 3(a)). This study chooses the original point that is the closest to the gravitational center as the relevant point (Fig. 3(b)), because selecting the closest point can effectively reducing the probability of producing errors.

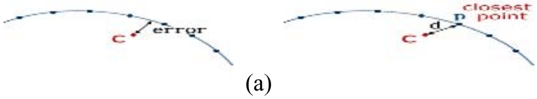


Fig. 3. (a) The gravitational center of cluster *c* is chosen as the relevant point; (b) *P* is chosen as the closest point to the gravitational center *c*, and set to the relevant point

2.3 Identify the Near Surface and Adjust the Appropriate Relevant Points

Another potential problem caused when space subdivision occurs on the near surface, revealing that the two surfaces are extremely close to each other. The near surface in Fig. 4(a) may lead to an incorrect judgment of flatness and cause non-manifold occurrences, because the respective points belonging to two surfaces may be merged into one relevant point (Fig. 4(b)). This mistake causes inconsistent curvature and errors in topology (Fig. 4(c)).

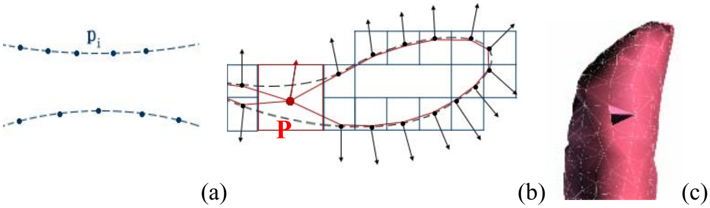


Fig. 4. (a) The near surface may lead to incorrect judgment; (b) Incorrect judgment for a near surface may cause non-manifold occurrence; (c) Reduction of the near surface to *p* causes inconsistent curvature and topology errors

The proposed near surface identification method has two parts. The first part comprises auxiliary points (u) that are used to detect the near surface inside the cluster. These auxiliary points are located at the cluster center c_i and the corners q_i (the centers of sub-clusters). For each auxiliary point u_i, the k closest points (p_k) are selected, and the cluster normal $\vec{e}_0(\vec{N})$ is calculated to obtain a reliable estimate of the signed inner product. The near surface can be easily distinguished according to the inner product sign ($\overrightarrow{p_i u_i} \bullet \vec{N}$), (Fig. 5(a)). The second part assumes that the nearest neighbor r of each point p_i in a cluster is found according to the value of its inner product ($\overrightarrow{p_i r_i} \bullet \overrightarrow{N}$) in order to detect the near surface (Fig. 5(b)). If a cluster may have a near surface, then it is subdivided to ensure that the near surface does occur (Fig. 5(c)).

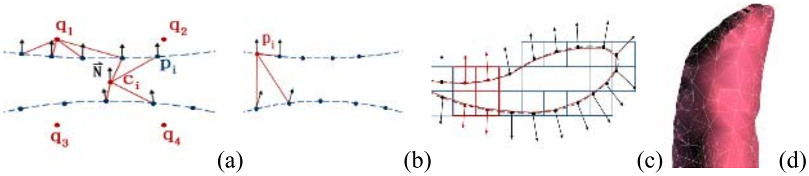


Fig. 5. (a) Auxiliary points at the cell center and corners are adopted to detect a near surface; (b) The inner product of the normal vector and adjacent points is adopted to determine whether a near surface exists; (c) The cluster containing a possible near surface is subdivided to avoid non-manifold occurrence; (d) Produced correct surface

2.4 Adaptively Add Dummy Vertices to Avoid Under-Sampling

Space subdivision can result in the obvious phenomenon of irregular density distribution between adjacent flat and feature areas. Unexpected holes during reconstruction resulting from under-sampling, due to insufficient information about for neighboring points in the local region (Fig. 6(a)). To avoid unexpected holes, the cluster based on the coplanar characteristic is again subdivided to produce dummy vertices, and increase and adjust the density of adjacent nodes. Therefore, information for adjacent points must be considered during reconstruction, and the levels for neighboring nodes must be restricted when constructing the tree structure. The nodes satisfying the coplanar restriction are affected by their neighboring sub-trees and subdivision continues.

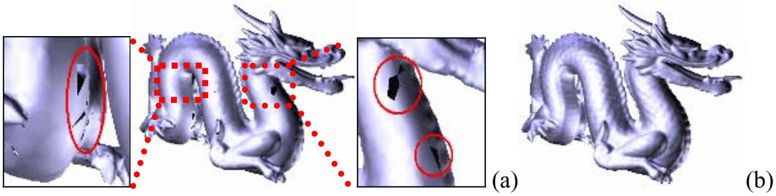


Fig. 6. (a) Unexpected holes during reconstruction resulting from under-sampling. (b) Reconstructed correct base model.

To avoid irregular density distribution and under-sampling caused by the coplanar restriction, subdivision continues until the level difference of adjacent clusters is within *n/2*, where *n* denotes maximum level of the octree even when the cluster is in accordance with the coplanar. The refined model is called the base model, and automatically represents different density distributions according to model variation. The density can be adjusted asymptotically at the points where different densities change, thus guaranteeing the accuracy of reconstruction. The experiment indicates that the number points in the base model are roughly *30*% of that in the original model, revealing that the cost of reconstructing the base model is *30*% of that of reconstructing the original model. This method ensures that the final triangular mesh is in accordance with object's topology (Fig. 6(b)). The subdivided dummy vertex then reduces to its original level in the following step.

2.5 Merge Dummy Vertex

Following correct reconstruction, the hierarchical tree-structure information can recover the base model simply and efficiently to a level based on the coplanar restriction. This efficient merging of a sub-tree brother reduces to its father node's location at the previous level. The experimental results indicate that using the coplanar variable *f =0.005* can yield a reconstructed model using roughly *10*% of the original points. Error rates for the reconstructed scheme, the scheme by QEM and the scheme by uniform subdivision pure space subdivision without restricting coplanar value) were compared (Fig. 7).

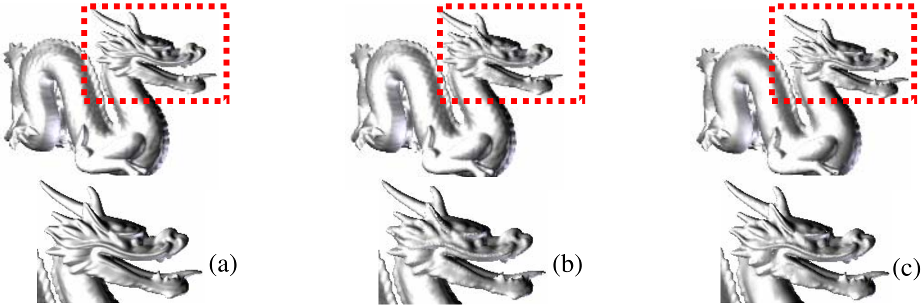


Fig. 7. The experimental results obtained by (a) QEM, (b) uniform subdivision and (c) the proposed method to simplify the same models with similar numbers of reconstructed model points. The QEM method achieves the best error rate, since it has mesh information; the proposed method adopts a coplanar restriction to simplify the original point cloud without additional information and to perform reconstruction, therefore maintaining a good error rate. Uniform subdivision has a regular distribution, but causes under-sampling in the feature area.

3 Hierarchical Simplification and Hierarchical Rendering for Multi-resolution Applications

The model can be displayed dynamically and quickly based on the octree data structure after reconstruction according to the coplanar restriction. Using the octree structure for space subdividing produces multi-resolution, and adjusting the octree level in various ways generates different display results. User-control dynamically adjusts the resolution level that can be displayed without further computations. As long as the octree data structure obtained by the previous calculation is adopted, sufficient simplified data can be provided to achieve rapid and effective simplification and update the connectivity information. The following methods can be adopted to obtain different rendering effects, where the user controls the number of relevant points.

Depth First reduction: The relevant points of a uniform distribution are produced. The nodes on the deepest level are first deleted and reduced to the relevant points of the previous level. Returning to the deepest level each time results in the simplification effect for a uniform distribution (Fig. 8(a)).

Reduction by one ring neighbor coplanar measurement: Each simplified relevant point can denote the surface information for each small region. The variation error denotes normal differences between adjacent relevant points, and can be adopted to estimate the coplanar degree of a relevant point for its adjacent region. Simplification operations are arranged in a priority queue according to the variation error ($Q*_{pqi}$) of each relevant point pair. The value of the coplanar and the relevant point of the selected point pairs are recalculated to ensure a good error rate (Fig. 8(b)).

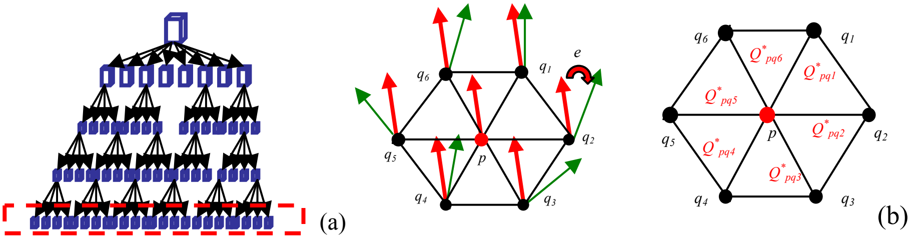


Fig. 8. (a) Depth First reduction reduces the deepest level each time; (b) Reduction by one ring neighbor coplanar measurement reduces the points according to the variation error of each relevant point pair $(Q^*_{pq}=Q_p+Q_q)$ $(Q_p=e_{pq1}+e_{pq2}+e_{pq3}+e_{pq4}+e_{pq5}+e_{pq6})$

4 Experimental Results

The following simplified models were obtained by the proposed method. This study confirms that the coplanar variable $f = 0.005$ obtains a good simplification result. The number of relevant points is slowly reduced (Fig. 9) when the coplanar variable f exceeds 0.005. Restated, 0.005 is an effective value for the degree of flatness in a model. The proposed algorithm adopts 0.005 as a default value, and can obtain a good error range (Table 1). The results of the proposed method are show in Figure 10 to Figure 12.

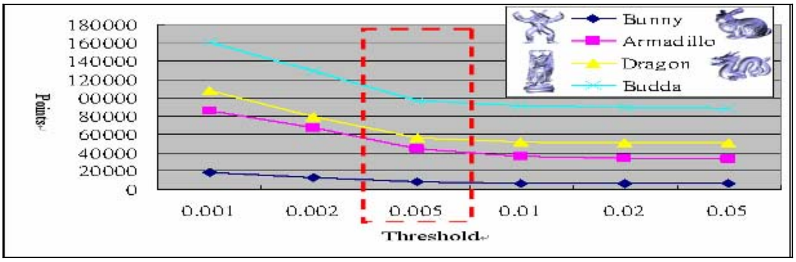


Fig. 9. The number of relevant points is slowly reduced when f exceeds 0.005

Table 1. The size generated by different models and simplified error measurement by Metro tool [11] and the flatness is using *0.005*

	Original points	base model	reconstructed model	Mean Error (absolute)	Mean Error (relative)
Dragon	437645	90374 (21%)	56302 (12.9%)	0.000058	0.000217306
Budda	543644	135205 (24%)	96967 (17.8%)	0.000032	0.000139719
Armadillo	172974	45077 (26%)	43981 (25.0%)	0.044266	0.00019346
Venus	134345	25479 (19%)	16276 (12.1%)	0.000083	0.000530694

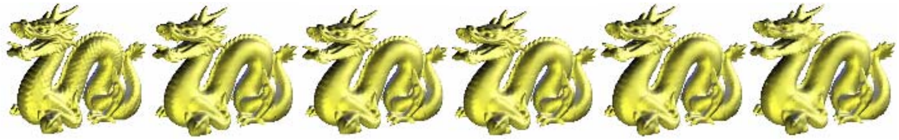


Fig. 10. The reconstructed results of the Dragon model on various levels after adopting the Depth First reduction. From left to right, Base model (*90374*points), reconstructed model (*56302* points), *55287* points, *49663* points, *30832* points, and *13237* points (*3*% of original).

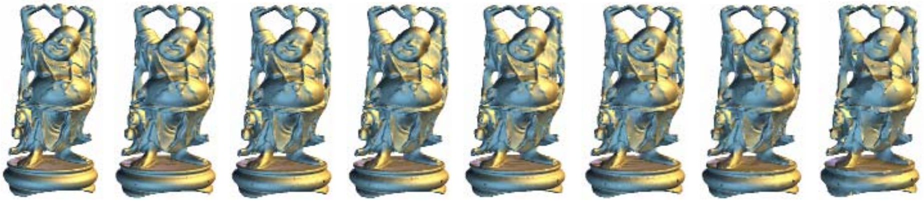


Fig. 11. The reconstructed results of the Budda model on various levels after using the one ring neighbor coplanar measurement. From left to right, Original model, Base model (*135205* points), reconstructed model (*96967* points), *70204* points, *50178* points, *40155* points, *20066* points; and *4963* points (*0.9*% of original).

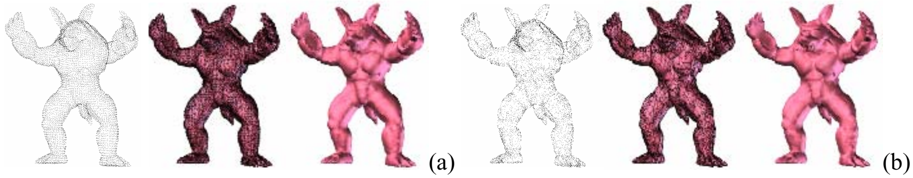


Fig. 12. The different point distribution of hierarchical rendering. (a) The depth first reduction obtains uniform distribution (11154 points) and (b) the one ring neighbor coplanar measurement product the feature-sensitive result (10014 points).

5 Conclusions and Future Work

This study presents a novel method the simplifying a point set using an octree structure to calculate the coplanar variable *f*, and spatially subdivide the sampled points in 3D. The input data only contains point coordinates. The final output includes a triangular mesh and octree data structure. Reducing the level of the octree can dynamically adjust its result without needing additional calculations. This proposed method facilitates producing a uniform and feature-sensitive simplified model for further mesh-based applications.

Further work will integrate point simplification and reconstruction algorithms; try to permit under-sampling and produce an appropriate simplified point set, and correctly reconstruct simplified point sets without increasing the dummy vertex.

References

1. A. Gu´eziec, " Surface simplification with variable tolerance", *Second Annual Intl. Symp. on Medical Robotics and Computer Assisted Surgery (MRCAS '95)* , 1995, pp.132–139.
2. B. Hamann, "A Data Reduction Scheme for Triangulated Surfaces", *Computer Aided Geometric Design*, Vol.11, 1994, pp. 197-214.
3. B. S. Jong, J.L. Tseng, W. H. Yang, T. and W. Lin, "Extracting Features and Simplifying Surfaces using Shape Operator", *The 2005 IEEE International Conference on Information, Communications and Signal Processing (ICICS 2005)*, 2005, pp. 1025-1029.
4. B. S. Jong, W. Y. Chung, P. F. Lee, and J. L. Tseng, "Efficient Surface Reconstruction Using Local Vertex Characteristics", *The 2005 International Conference on Imaging Science, Systems, and Technology : Computer Graphics*, 2005, pp. 62-68.
5. C. Moenning and N. A. Dodgson, "Intrinsic point cloud simplification", *In Proc. 14th GrahiCon*, Vol. 14, 2004.
6. J.D. Boissonnat and F. Cazals, "Coarse-to-fine surface simplification with geometric guarantees", *EUROGRAPHICS'01, Conf. Proc*, 2001, pp. 490–499.
7. J. Rossignac and P. Borrel, "Multi- resolution 3D approximations for rendering complex scenes", *Modeling in Computer Graphics: Methods and Applications*, 1993, pp. 455–465.
8. M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and T. Silva, "Point Set Surfaces", *In Proc. 12th IEEE Visualization Conf.*, 2001, pp. 21–28.
9. M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics", *SIGGRAPH 97 Conference Proceedings*, 1997, pp. 209–216.
10. M. Pauly, M. Gross and L. P. Kobbelt, "Efficient Simplification of Point-Sampled Surfaces", *In Proc. 13th IEEE Visualization Conf.*, 2002, pp. 163-170.
11. P. Cignoni, C. Montani, and R. Scopigno, " Metro: Measuring Error on Simplified Surfaces" , *Computer Graphics Forum*, Vol.17 , No.2 , 1998, pp.167-174.
12. R. All`egre, R. Chaine, and S. Akkouche, "Convection-driven dynamic surface reconstruction", *In Proc. Shape Modeling International*, 2005, pp. 33–42.
13. T. K. Dey, J. Giesen and J. Hudson, "Decimating Samples for Mesh Simplification", *In Proc. 13th Canadian Conference on Computational Geometry*, 2001, pp. 85–88.
14. T. S. Gieng, Bernd Hamann, Kenneth I. Joy, Gregory L. Schussman, and Issac J. Trotts, "Constructing hierarchies for triangle meshes", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 4(2) , 1998, pp.145–161.

Dimension Reduction in 3D Gesture Recognition Using Meshless Parameterization

Yunli Lee, Dongwuk Kyoung, Eunjung Han, and Keechul Jung

HCI Lab., School of Media, College of Information Technology,
Soongsil University, Seoul, South Korea
{yunli, kiki227, hanej, kcjung}@ssu.ac.kr
http://hci.ssu.ac.kr

Abstract. 3D gesture recognition offers more details data but leads to computational hurdles which do not support real-time gesture recognition application. In this paper, we introduce a method of dimension reduction for 3D gesture recognition. Our method uses meshless parameterization to perform dimension reduction in modeling process and extracts gesture data, in order to reduce the computation complexity. In addition, this method also maintains the 3D gesture information and result novel features vectors for 3D gesture recognition. The computational efficiency of dimension reduction and by using novel features vectors makes 3D gesture recognition more possible to achieve real-time performance.

Keywords: Dimension reduction, 3D gesture recognition, gesture data, meshless parameterization.

1 Introduction

Gesture is a meaningful form of body motions used in daily life as a means of communication. We wave to stop the taxi, we nod our head when agree, we raise a hand to get attention, or even we point at things we want. These all are called gesture. Human body can express a variety of gestures to communicate.

Since computer usages in our daily lives rapidly increase, gesture is becoming popular for interaction media in human computer interaction (HCI) environment to provide more natural and intuitive interaction among people and computer. The traditional interaction media such as keyboard or mouse is not well suited and inherently limit the speed and naturalness for interaction. Thus, novel devices and techniques of gesture recognition have emerged over the past two decades [4 – 17].

Gesture recognition is a process to identify user's gestures and the system responds to them accordingly. The first attempts to solve the problem resulted in developing mechanical devices that allow gestures to be used as a form of input for HCI. However the use of mechanical devices such as a special data glove [6] requires the user to wear a cumbersome device and the cost is expensive therefore vision-based techniques have been brought into use and provide more naturalness and immersive interaction in HCI environment.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 64 – 73, 2006.

Till now, most researches have focused on 2D representation gesture recognition. The main reason for this bias towards 2D gesture recognition was because the data are simple, sufficient for gesture recognition and reasonably computational processing time [8, 13, 14]. However currently 3D gesture recognition is a challenging task, where it gives more accurate features data such as human body orientation and shape. However, the 3D representation recognition leads to expensive computational processing time and complexity of features extraction [7, 15].

There are few problems must be considered in the process of implementing the gesture recognition system. The first process is choosing the gesture model where the model may consider spatial and temporal characteristics of human actions. The second process is gesture data extraction to determine the type of data representation. The third process is gesture recognition, where the gesture parameters are classified and interpreted [9, 17]. Definition and selection of the gesture data are one of the problems in vision-based recognition, hence, it attracts our interest and some related works are presented in Section 2.

This paper only focuses on two processes – gesture modeling and gesture data extraction (see Fig. 1), and we believe these are the preliminary and more important than other processes. Although 2D representation is simple and reasonably computational processing time, however, it faces difficulties for estimating human posture due to lost information caused by self-occlusion and image projection [16]. Therefore, we choose 3D representation model as a form of input gestures for gesture modeling, which offers more details of modeling but leads to computational hurdles that do not support real-time application. Therefore, this paper aims to overcome the computation complexity and preserve the 3D gesture for recognition. As we use a meshless parameterization method [2] for gesture modeling and gesture data extraction, we can get a reasonable processing speed and preserve the 3D features for 3D gesture recognition.

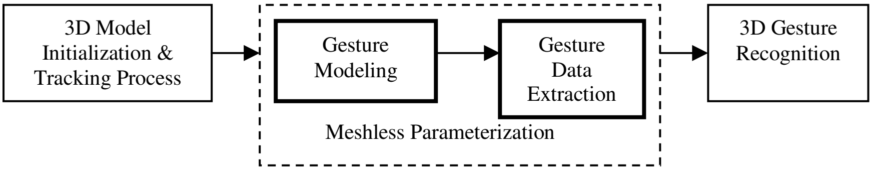

3D Model Initialization & Tracking Process
Gesture Modeling
Gesture Data Extraction
Meshless Parameterization
3D Gesture Recognition

Fig. 1. Overview of gesture recognition process

Generally, meshless parameterization is defined as a one-to-one mapping process into some convex parameter domain in the plane. This method provides a simple, fast and robust ways, yields good results in numerical tests and triangulations with better shaped triangles [1, 2, 3]. The meshless parameterization of 3D surface is commonly used for texture mapping, morphing, surface fitting and etc. However, we use this method for reconstructing the 3D volume data into some convex parameter of gesture data representation, which results an image-based representation.

We discuss how to reduce dimension of 3D volume data using meshless parameterization algorithm in Section 3. In Section 4 we present novel features that resulted from meshless parameterization for 3D gesture recognition. The experimental results and novel gesture features are showed in Section 5. Some thoughts about future work and conclusion are remarks in Section 6.

2 Related Work

In this section, we discuss about some approaches that had been proposed for 3D gesture recognition of vision-based. The most straightforward one is using multiple cameras to acquire visual information of human in some specify environment and try to extract the necessary gestures. However, this approach faces several difficulties in the process of features extraction for 3D gesture recognition, such as a choice for feature vectors computed from the 3D gesture data output by the vision system, hindered by complexity of feature extraction and model parameter estimation. Most features vectors are generated by computing the absolute and relative ellipse positions, orientations, velocities and accelerations.

Lee W. Campbell et al. [11] study on ten different features vectors for 3D gesture recognition. A set of 18 T'ai Chi gestures recognition performance are tested and compared the performance of ten different feature vectors based on 3D hand and head tracking data. From the results, they highlight several important issues associated with the general problem of choosing features for gesture recognition systems. The right set of choosing features affect recognition performance. Therefore, a careful consideration of features extraction design can lead to significantly improved results.

The 3D gesture data representations are divided into two types: object-based or model-based representation like point, box, silhouettes or blob, and image-based or appearance-based representation like spatial, spatio-temporal, edge or features [17].

With object-based approach [5, 10, 11], it is possible to capture human gesture in high dimensionality than 2D. However, it is too complex to be rendered in real-time and reconstruct the 3D model. S. Malassiotis et al. [5] using 3D sensor to generate range data for gesture recognition. The system improved efficiency and robustness through 3D information. However, the usage of special camera or 3D sensor, and the difficulty and computational complexity of visual 3D localization and robust tracking leads us toward other possible approach for gesture recognition.

H.K. Shin and et al. [10] proposed 3D Motion History Model (MHM) for gesture recognition. The method using stereo input sequences which contain motion history information in 3D space and overcome the 2D motion limitation like viewpoint and scalability. Nevertheless, Motion History Image (MHI) is more easy and fast algorithm. Therefore, 3D appearance or view-based representation is widely used for 3D gesture recognition [4, 7]. R. Fablet and M.J. Black proposed automatic detection and tracking of human motion with view-based representation. They developed a novel representation of human motion using low-dimensional spatio-temporal models that are learned using Principal Component Analysis (PCA) [4].

Guangqi Ye and et al. [7] present 3D gesture recognition scheme that combines the 3D appearance and motion features. They reduce the dimensionally of the 3D features by employing unsupervised learning. The proposed method is flexible and efficient way to capture the 3D visual cues in a local neighborhood around the object.

The main difference between our approach and the existing methods is that we propose a method to reduce dimension of object-based into image-based representation using meshless parameterization, without loss of the 3D shape generality and information. Then, we introduce a novel feature vector that extracted from the image-based representation, which is a reasonable feature vector for 3D gesture recognition.

3 Gesture Modeling: Meshless Parameterization

3D model representation is complex and causes difficulties for gesture data extraction and parameter estimation. Therefore, for gesture modeling process, instead of 3D model triangulation, we represent the model into a 2D representation. Parameterization of 3D model without mesh information into some convex parameter of 2D representation is called meshless parameterization [1, 2]. The meshless parameterization determines a sequence of parameter points from the 3D model without any given structure topology. The method works well on surface patch with one boundary. Formerly, the bottom part of 3D human body is rarely used for gesture recognition. Therefore, we choose only the upper part of human body for gesture model that sampled on single surface patch with one boundary.

The upper part of human body is defined as open surface S in R^3, which means a single surface patch with one boundary. Set a domain D in a unit square, as planar in R^2, we use meshless parameterization method to find one to one mapping function, $F: S \rightarrow D$, where the 3D gesture points of $S\ (x,y,z)$ is mapped into 2D pixels of $D\ (u,v)$.

The basic idea of meshless parameterization for dimension reduction is presented in this paper. This method is required to determine the boundary points and appropriate choice of radius for ball neighborhood. First, we assume the 3D gesture points as a set of P. This set of P has N points, it consists of two disjoint subsets, $P_I=\{p_1,p_2,\ldots,p_n\}$ as a set of interior points with n points, and $P_B=\{p_{n+1},p_{n+2},\ldots,p_N\}$ as a set of boundary points with N-n points. The boundary points need to be in ordered sequence before mapping. The algorithm for splitting the set P into two disjoints subsets and maps the boundary points into the domain D, explain in next section.

Meshless parameterization method involves two basic steps. The first step is to map the boundary points P_B into the boundary of domain D plane. Then, the corresponding parameter points $U = \{u_{n+1},\ u_{n+2},\ldots,u_N\}$ are laid around the domain D in counter-clockwise order. The distribution of parameter points U are based on the chord length parameterization.

The second step, we map the interior points into the domain D plane. However, before mapping, we have to choose a neighborhood p_j for each interior point in P_I where the points are some sense close by, and let N_i as a set of neighborhood points of p_i. In this case, we choose a constant radius r, by computing average distance of two boundary points, for a ball neighborhood. The points that fall within the ball are considered the neighborhood points. Fig. 2 shows a ball neighborhood with constant radius r.

Then, we compute the weights for the point to the corresponding neighborhood points using the reciprocal distance weight, which means that weights of neighborhood points must be equal to one, to guarantee the convex combination. To get the interior points $U = \{u_1,\ u_2,\ldots,u_n\}$ of the corresponding point P_I, the linear system of n equations are solved.

3.1 Boundary Splitting and Ordering

In our approach, the set of P consists of unordered vertices sequences. In order to split the set of P into boundary set P_B, and interior set P_I, we assume the model is open surface which consist of only one boundary. The 3D model volume is given x-axis and y-axis as base plane, and z-axis is referred to the height of the model. Therefore, we can determine the model boundary points using the z-axis information. As a result, we get a point with minimum value of z-axis, then search for the point that has same minimum value of z-axis, set the point as boundary and ordered these points using boundary-following algorithm. These boundary points are mapped in counter-clockwise order onto the domain D plane. Each parameter point is computed based on chord length parameterization.

The basic algorithm for splitting data set and mapping boundary points can be described as follow:

1. *Cut the human body into two parts, the cutting edge of upper part's points are set as boundary points*
2. *Order the boundary points using boundary-following algorithm*
3. *Get the first point from boundary set and place it to the 2D domain origin*
4. *Follow by the next boundary point, compute the chord length between the previous point and the current point, then map the point to the 2D domain in counter-clockwise order*
5. *Repeat the step 4 until all boundary points are mapped to the 2D domain*

3.2 Neighborhoods, Weights and Linear System

In this section, we refer to Floater and Reimers [2] methods for choosing the neighborhoods *Ni* and weights λ_{ij} for each interior point *pi*. A ball neighborhood is used to determine set of *Ni* for neighborhood points of *pi* (see Fig. 2). Then, the reciprocal distance weights method is to compute the weight λ_{ij} for each interior point *pi*. To determine the radius *r* of ball neighborhood, we set the radius *r* constant. The equation below explains the theory for choosing neighborhoods and weights.

$$\begin{aligned}
&\text{Let } N_i \text{ be the ball neighborhood} && (1)\\
&N_i = \{j : 0 < \|p_j - p_i\| < r\},\\
&\text{for some radius } r > 0\\
&\text{and let the } \lambda_{ij} \text{ be the reciprocal distance weights}\\
&\lambda_{ij} = \frac{1}{\|p_j - p_i\|} \Big/ \sum_{k \in N_i} \frac{1}{\|p_k - p_i\|} .
\end{aligned}$$

The choice of weights is positive λ_{ij}, for $j \in N_i$, such that $\sum_{j \in N_i} \lambda_{ij} = 1$ where the parameter interior point, u_i is some convex combination of its neighbor's u_j. In order to compute the n parameter points $U=\{u_1,u_2,\ldots,u_n\}$ for the corresponding to the interior point p_i, we solve the linear system of n equations:

$$u_i = \sum_{j \in N_i} \lambda_{ij} u_j, \quad i = 1, \ldots, n \;. \tag{2}$$

From above equation, we rewrite the linear system in the form of *Au*=*b*, where *A* is a matrix of weight nxn, *u* is parameter points and *b* is the sum of neighbor points of *u*. The linear system could be express in the matrix form,

$$\text{where } A = \begin{bmatrix} 1 & -\lambda_{12} & \cdots & -\lambda_{1n} \\ -\lambda_{21} & 1 & \cdots & -\lambda_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\lambda_{n1} & -\lambda_{n2} & \cdots & 1 \end{bmatrix}, \; a_{ij} = \begin{cases} 1 & , \quad j = i \\ -\lambda_{ij} & , \quad j \in N_i \\ 0 & , otherwise \end{cases} \tag{3}$$

$$\text{and } u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, \; b = \begin{bmatrix} \sum_{j \in N_1} \lambda_{1j} u_j \\ \sum_{j \in N_2} \lambda_{2j} u_j \\ \vdots \\ \sum_{j \in N_n} \lambda_{nj} u_j \end{bmatrix}.$$

We use Gauss Elimination to compute the inverse of matrix *A*. Then, the equation (2) is solved and obtained the parametric value *U* for all interior points, which are used to represent gesture data, and novel features are extracted for gesture recognition.

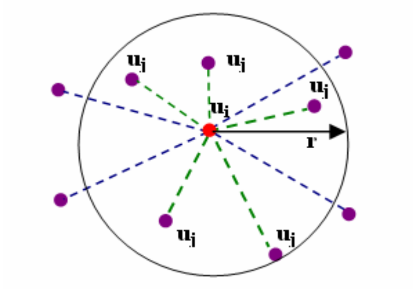


Fig. 2. A ball neighborhood with constant radius *r* and total the number of neighborhoods $N_i\{u_j\}$=5 for u_i

4 Gesture Data Extraction

Definition and selection of gesture data are important because it greatly affects the recognition performance. Thus, we are focusing on how to choose a good feature for recognition.

This paper use meshless parameterization for gesture modeling, which reduces the dimensional complexity and maintains most information of 3D without loss of the shape generality. Our method extracts the gesture data after the dimension reduction using meshless parameterization. The method results novel features vectors like pixel location *(u_i, v_i)*, pixels distribution based on the interior points and number of local neighborhoods for each pixel, which refer to a ball neighborhood of each interior points, these features vectors make 3D gesture recognition is possible to achieve real-time performance.

Our proposed features are totally different from the existing features like feature moments, orientation, areas, which need additional process for extracting. The proposed features vectors are computed during the process of dimension reduction using meshless parameterization, therefore the computational times are reduced. The idea of gesture features extraction and matching are shown in our experimental results, in section 5.

5 Experimental Results

Initially, we tested our proposed method to an artificial data set to show our principal idea. The artificial open cube data set consists of 89 gesture data points. Fig. 3(a) shows the 3D representation of open cube as input. Fig. 3(b) shows the boundary points are detected. Here, the boundary points are located at minimum value of z-axis, which shown by connected lines. After the boundary points are detected, these points are map into the 2D unit square domain using the chord length parameterization in Fig. 3(c). The Fig. 3(d) shows the result of meshless parameterization, where the interior gesture data points map into the 2D unit square domain, which result an image-based representation of 3D gesture data.

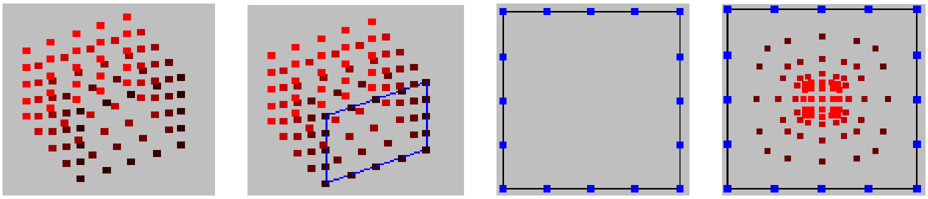


Fig. 3. (a) 3D volume of open cube; (b) boundary points are detected; (c) mapped boundary points on 2D unit square domain; (d) mapped the interior points on 2D unit square domain

The open cube in Fig. 3(a) with some deformation is assumed as a gesture. Fig. 4(a) shows initial position of the open cube model in image-based representation, which introduces some novel features like pixel location *(u_i, v_i)*, pixels distribution and number of local neighborhoods for each pixel. Fig. 4(b) shows image-based representation of open cube after deformed, where the points of top surface are moved downwards, notice that the most internal pixels are represented the top surface points. From this experiment result, the features are extracted from the image-based representation and shown reasonable features for gesture recognition. In Fig. 4(c) shows another example, the both corners on the top front surface were deformed, the features are extracted for matching, shown in small rectangle window size. This image-based representation consists of 3D gesture information, where each pattern distribution features is unique and possible for matching and recognition.

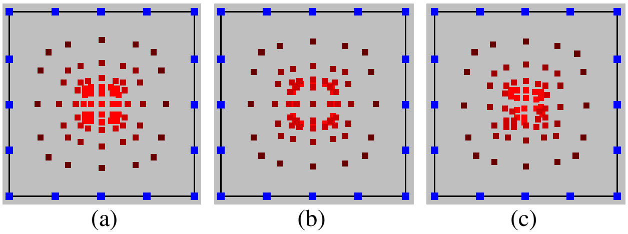


Fig. 4. (a) Image-based representation of initial posture; (b) top surface deformation gesture; (c) front top corners deformation gesture

Further more, we tested on actual human data sets, downloadable at http://hci.ssu.ac.kr/yllee_research.html, to show our method works well in human model. Fig.5 (a) shows the capturing system architecture with four cameras setting within the box volume. The human model is located within the box volume are captured and generate a 3D volume data set Fig.5 (b). Then, our propose method is used to reduce the dimension of this 3D volume data set into image-based representation.

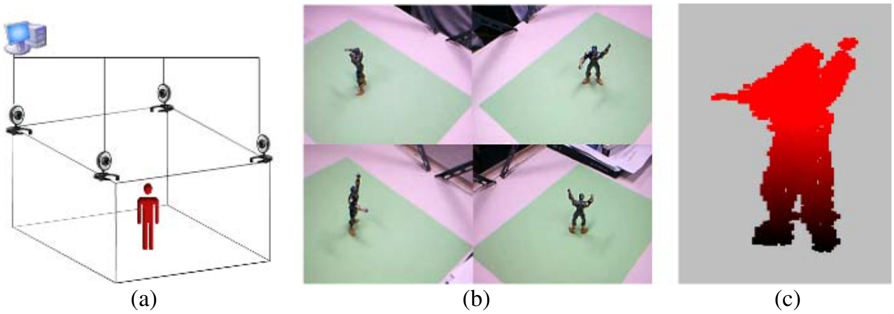
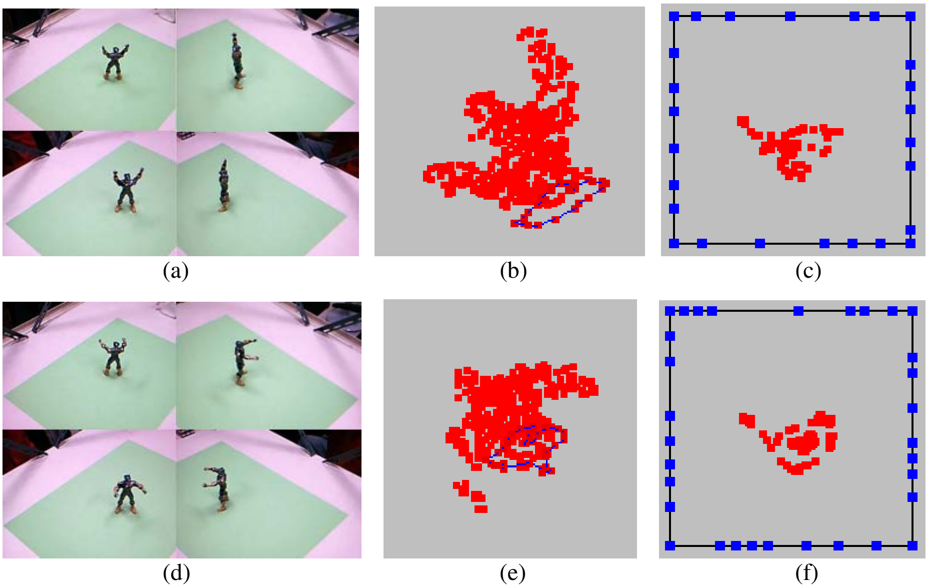


Fig. 5. (a) The capturing system architecture, four cameras set within the box volume; (b) four images captured from the system of human gesture posture; (c) 3D volume data set

The results in Fig.6.(c) and Fig.6.(f) represent two different gestures in image-based representation. We can compare both gestures using extracted features vectors like number of neighborhoods for each interior gesture data. Each gesture image-based representation has unique pattern distribution. Therefore, it is reasonable for gesture recognition and matching purpose.


(a) (b) (c)

(d) (e) (f)

Fig. 6. (a) and (d) show four images captured from the system with corresponding gesture; (b) 3D volume of upper body for (a) gesture, where both hands raise vertically toward top position; (c) image-based representation of (b) volume data; (e) 3D volume of upper body for (d) gesture, where both hands raise horizontally toward front position; (f) image-based representation of (e) volume data

6 Conclusions and Future Work

Our primary aim is to reduce the 3D volume complexity without loss of the information generality and satisfy the critical requirements of speed and robustness in features extraction for 3D gesture recognition. This paper has described an approach includes possible main steps in dimension reduction for 3D model and extract novel features for 3D gesture recognition. The simple, fast and efficient of meshless parameterization method leads the idea to parameterize the gesture model and features extraction. In addition, this method map 3D volume data by one to one into 2D domain plane, thus most 3D features are maintained in image-based representation.

Our experimental results show potential of meshless parameterization for gesture modeling and features extraction of 3D model. The method reduces the model dimension without loss of data generality or maintains 3D gesture information. The number of neighbor points, location and pattern distribution are extracted as gesture features vectors. These features are used for matching and recognizing purpose. Base on the results of feature extraction, we will continue to implement the gesture recognition system and develop more gestures data for testing. Further, we also plan to construct automatic features extraction for interested part instead of the whole model to achieve better performance and recognition accuracy.

Acknowledgments. This work was supported by grant No. (R01-2006-000-11214-0) from the Basic Research Program of the Korea Science & Engineering Foundation.

References

1. Floater M. S.: Meshless Parameterization and B-spline Surface Approximation. The Mathematics of Surfaces IX, R. Cipolla and R. Martin (eds.). Springer-Verlag (2000) 1-18
2. Floater M. S., Reimers M.: Meshless Parameterization and Surface Reconstruction. Computer Aided Geometric Design (2001) 77-92
3. Floater M. S., Hormann K.: Surface Parameterization: a Tutorial and Survey. Advances in Multiresolution for Geometric Modelling (2004) 157-186
4. R.Fablet, M.J. Black: Automatic Detection and Tracking of Human Motion with a View-Based Representation. 7th European Conference on Computer Vision, Copenhagen, Denmark, Proceedings, Part 1 (2002) 476
5. Malassiotis S., Aifanti N., Strintizis M.G.: A Gesture Recognition System Using 3D Data. Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (2002) 190-193
6. Thomas S. Huang, Vladimir I. Pavlovic: Hand Gesture Modeling, Analysis, and Synthesis. Int. Workshop on Automatic Face-and Gesture-Recognition, Zurich (1995) 26-28
7. Guangqi Ye, Jason J. Corso, Gregory D. Hager: Gesture Recognition Using 3D Appearance and Motion Features. Proceeding IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (2004)
8. Qiulei Dong, Yihong Wu, Zhanyi Hu: Gesture Recognition Using Quadratic Curves. ACCV, LNCS 3851 (2006) 817-825
9. Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7 (1997)
10. Ho-Kuen Shin, Sang-Woong Lee, Seong-Whan Lee: Real-Time Gesture Recognition Using 3D Motion History Model. ICIC 2005, Part I, LNCS 3644 (2005) 888 – 898
11. Lee W. Campbell, David A. Becker, Ali Azarbayejani, Aaron F. Bobick, Alex Pentland. Invariant Features for 3-D Gesture Recognition. M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 379, Second International Workshop on Face and Gesture Recognition, Killington V.T (1996)
12. Huang Yu, Xu Guang-You, Zhu Yuan-Xin: Extraction of Spatial-Temporal Features for Vison-Based Gesture Recognition. J. Computer Science and Technology (2000) 64-72
13. Mori G., Xiaofeng Ren, Alexei A. Efros, Jitendra Malik: Recovering Human Body Configurations: Combining Segmentation and Recognition. CVRP'04, Volume 2, Washington, DC (2004) 326-333
14. Pengyu Hong, Matthew Turk, Thomas S. Huang: Gesture Modeling and Recognition Using Finite State Machines. IEEE Conference on Face and Gesture Recognition (2000)
15. Yoichi Sato, Makiki Saito, Hideki Koike: Real-time Input of 3D Pose and Gestures of a User's Hand and Its Applications for HCI. Proceeding IEEE Virtual Reality Conference (2001) 79-86
16. Chi-Wei Chu, Isaac COHEN: Posture and Gesture Recognition using 3D Body Shapes Decomposition. IEEE Workshop on Vision for Human-Computer Interaction (2005)
17. Thomas B. Moeslund, Erik Granum: A Survey of Computer Vision-Based Human Motion Capture. Computer Vision and Image Understanding (2001) 231-268

Target Calibration and Tracking Using Conformal Geometric Algebra

Yilan Zhao[1], Robert Valkenburg[2], Reinhard Klette[1], and Bodo Rosenhahn[3]

[1] Computer Science Department, The University of Auckland, New Zealand
[2] Industrial Research Limited, Auckland, New Zealand
[3] Max Planck Institute, Saarbrücken, Germany

Abstract. This paper is about real-time refinement of the 3D positions of a large number of stationary point-targets from a sequence of 2D images which are taken by a hand-held, calibrated camera group. To cope with the large data quantity arriving rapidly, an efficient iterative algorithm was developed. The problem and solution are expressed entirely within the computational framework of conformal geometric algebra. The iterative solution requires a pose estimation step of which two strategies are investigated. Experiments are performed to evaluate the algorithm based on synthetic and real data.

Keywords: conformal geometric algebra, pose estimation.

1 Introduction

Recovering the positions of many point-targets over a large area is computationally expensive. This paper describes an efficient iterative algorithm to refine target positions from a sequence of 2D images. The targets used in this project are point-lights (left Figure 1) and form part of a flexible 6D positioning system. A group of rigidly co-located calibrated cameras (right Figure 1) is moved along an arbitrary path and takes images of the targets. The image points of the targets are transformed to 3D lines which are used by the algorithm to update the 3D positions of the targets. The algorithm is expressed entirely within the computational framework of conformal geometric algebra (CGA). The previously developed target calibration algorithm described in [9] is non-iterative and requires all the line data to be gathered before the algorithm can proceed. It can be used to obtain an initial estimate of the target positions for the iterative algorithm described in this paper. This work is a continuation of work reported in [9] in the application of the conformal model of geometric algebra.

1.1 Geometric Algebra and Conformal Model

In this section, the basic concepts and operations of geometric algebra that are required in this paper are briefly introduced. For a detailed introduction to geometric algebra, refer elsewhere e.g. [1,2,3].

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 74–83, 2006.

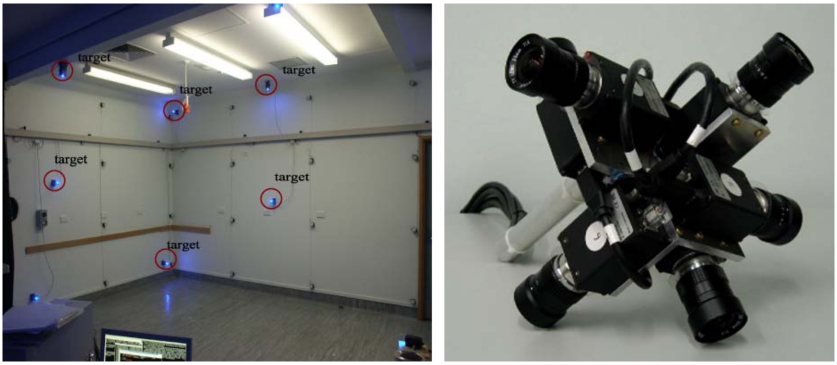


Fig. 1. Left: targets; six of them are encircled. Right: camera group.

Geometric algebra (GA) is the application of Clifford algebras to geometric problems. It integrates many concepts and techniques, such as linear algebra, vector calculus, differential geometry, complex numbers and quaternions, into a coherent framework. A geometric algebra over $\mathbb{R}$ is denoted $\mathcal{G}_{p,q}$ with p positive and q negative basis elements. Let x_1, x_2, ..., x_r be vectors. $X = x_1 \wedge x_2 \wedge \ldots \wedge x_r$ is referred to as an r-blade where '$\wedge$' is called *outer product*. r is the *grade* which indicates the dimensionality of the blade. A linear combination of multiple r-blades constructs an *r-vector*. $\mathcal{G}^r_{p,q}$ denotes the r-vectors in $\mathcal{G}_{p,q}$. A linear combination of a set of elements with different grades is a *multivector*. For example, if A is a multivector then it can be written as $A = \sum_r \langle A \rangle_r$ where $\langle A \rangle_r$ represents the grade r part of A. $\langle A \rangle$ or $\langle A \rangle_0$ represents the scalar part of A. The part of A containing the grades in another multivector B is denoted as $\langle A \rangle_B$. $A \rfloor B = \Sigma_{r,s} \langle \langle A \rangle_r \langle B \rangle_s \rangle_{s-r}$ is defined as the left contract inner product of A and B. The outer product can be related with the inner product by the following equation: $A \rfloor (B \rfloor C) = (A \wedge B) \rfloor C$. *Reverse* of X is defined as $\widetilde{X} = x_r \wedge \ldots \wedge x_2 \wedge x_1$. The dual of a blade X is defined as $X^* = X \rfloor I^{-1}$, where the pseudo-scalar I is an n-blade $e_1 \wedge \ldots \wedge e_n$ based the orthogonal basis ($\{e_i : \ i = 1 \ldots n\}$, $e_i \cdot e_j = 0$ for $i \neq j$, $e_i \cdot e_i = 1$) of $\mathbb{R}^n$ within $\mathcal{G}_n$. The norm of a multivector A can be calculated by $|A| = \sqrt{\left| \left\langle \widetilde{A} A \right\rangle \right|}$. If S is a linear operator, the *outermorphism* $\underline{S}$ is defined by $\underline{S}(X) = S(x_1) \wedge S(x_2) \ldots \wedge S(x_r)$. The derivative of multivector valued function F with respect to multivector X is denoted $\partial_X F$. $\dot{\partial}_X F \dot{G}$ means differentiate $G = G(X)$ with respect to X while regarding F as a constant. The following result [10] is required in later developments,

$$\partial_X \left\langle X Y X^{-1} Z \right\rangle = \left\langle Y X^{-1} Z \right\rangle_X - \left\langle X^{-1} Z X Y X^{-1} \right\rangle_X$$

where X, Y, Z be multivectors where Y and Z are independent of X.

GA expresses a number of models of 3D Euclidean space ($\mathcal{E}^3$), such as 3D Euclidean model, 4D homogeneous model and 5D conformal model. In this paper we use the conformal model of geometric algebra (CGA) based on $\mathcal{G}_{4,1}$. $\mathcal{G}_{4,1}$ is based on the orthonormal basis $\{e_1, e_2, e_3, e_+, e_-\}$ where $e_k^2 = e_+^2 = 1$ and $e_-^2 = -1$. It is usually more convenient to use the basis $\{e_o, e_1, e_2, e_3, e\}$ as it has a better geometric interpretation, where $e_o = \frac{e_- - e_+}{2}$ is associated with the origin and $e = e_- + e_+$ with the point at infinity. CGA allows a diversity of objects to be represented directly as blades (e.g. point, line, plane, circle, sphere, tangents and orientations) and allows a variety of operations to be represented as versors (e.g. rotor, translator, motor). A vector is represented as $v = v_1 e_1 + v_2 e_2 + v_3 e_3$ where v_1, v_2, v_3 are scalars. A point with location at the Euclidean point $\boldsymbol{p} \in \mathcal{G}_3^1$ is represented as $p = \boldsymbol{p} + e_o + \frac{1}{2}\boldsymbol{p}^{\,2} e \in \mathcal{G}_{4,1}^1$. A line is represented by $\Lambda = p \wedge v \wedge e$ where $p \in \mathcal{G}_{4,1}^1$ is a point and $v \in \mathcal{G}_3^1$ is a direction vector. A line is normalised by the mapping $\Lambda \rightarrow \frac{\Lambda}{\|\Lambda\|}$. A dual sphere centered at point p with radius ρ is given by $s = p - \frac{1}{2}\rho^2 e$. A Euclidean motion is represented by a *motor* $M = \exp\left(-\frac{1}{2}B\right)$ where $B = \boldsymbol{B} - \boldsymbol{t}e$ where $\boldsymbol{B} \in \mathcal{G}_3^2$ and $\boldsymbol{t} \in \mathcal{G}_3^1$. A motor M has properties which are important for deriving the algorithm: (i) $M \in \mathcal{G}_{4,1}^{0,2,4}$, (ii) $M\widetilde{M} = 1$, (iv) if $X \in \mathcal{G}_{4,1}^k$ then the transformation of X is given by $MX\widetilde{M} \in \mathcal{G}_{4,1}^k$.

1.2 Problem Description

The targets are defined in a world coordinate system denoted by CSW. Since the geometric relationship between the individual cameras which comprise the camera group is fixed and known, the camera group can be associated with a single moving coordinate system denoted by CSM.

An initial estimate of the positions of n targets $\{p_i^0 \in \mathcal{G}_{4,1}^1, i = 1 \dots n\}$ is given [9]. The initial pose of the camera group CSM is also given and represented as a motor M_o. The camera group CSM is moved to m positions on the path in CSW. The movement of CSM is tracked and represented by a sequence of motors $M_k, k = 1 \dots m$. At each position in CSW, a set of images are captured and the image points of the targets are extracted and converted to normalised lines $\{\Lambda_i^k \in \mathcal{G}_{4,1}^3, i = 1 \dots n, k = 1 \dots m\}$ in CSM. These lines are processed to refine the initial target position estimates. When CSM is moved to the next position, the new estimate of target positions will be calculated based on the previous estimate and a new set of lines. For m positions on the path, m updates are performed.

The problem can now be summarised as follows: Given a group of lines in CSM, a previous estimate of a set of points and a previous pose, we wish to update the coordinates of these points in CSW.

2 Target Refinement Using Geometric Algebra

The solution to the problem is analysed and developed in this section. At the beginning of the motion of the camera group we are given initial positions of targets and the initial pose of CSM. At each position we are given a new set of lines between optical centers and visible targets in CSM. The following steps need to be done during camera motion: (i) pose estimation of CSM; (ii) transformation of corresponding lines from CSM into CSW; (iii) update of target positions.

2.1 Pose Estimation: Objective Function Versus Point-Line Constraint

We estimate the pose of CSM by two alternative iterative strategies (i) non-linear optimisation of an objective "error" function. (ii) root finding of a 4-blade *point-line constraint* equation.

Non-linear optimisation of an objective function. The distance d between a point p and a line Λ is defined [10] by $d^2(p, \Lambda) = -\frac{1}{2}\langle \Lambda p \Lambda p \rangle$. The total distance between all points and their associated lines is defined as follows:

$$d^2 = \sum_j \sum_i \alpha_i \left(d^2(p_i, \Lambda_j) \right) \tag{1}$$

where $\alpha_i \in \{0, 1\}$ indicates whether the target is visible by any of the cameras. p_i is a target point and Λ_j is assumed to be a line which connects p_i to different cameras (i.e., their optical centers) in CSW. If the lines are given in CSM and the pose of CSM is represented by M then Λ in Equation (1) is replaced by $M\Lambda\widetilde{M}$ giving

$$d^2(M) = -\frac{1}{2} \sum_i \sum_j \alpha_i \left\langle (M\Lambda_j\widetilde{M}) p_i (M\Lambda_j\widetilde{M}) p_i \right\rangle \tag{2}$$

This objective function produces a scalar with a well-defined geometric meaning.

The poses of CSM are estimated using a Quasi-Newton optimization technique which is described in [6] (pages 425–430). We use a non-linear minimisation routine (called "dfpmin") which implements the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) update.

The optimization routine requires an objective function and its gradient. The motor M representing the pose of CSM is parameterised $M = M(x)$ where $x \in \mathbb{R}^6$. We use $M(x)$ in the objective function d^2 in Equation (2) to express the objective function as $g(x) = d^2(M(x))$. The gradient is given by $[\nabla_x g(x))]_i = \partial_{x_i} g(x) = \partial_{x_i} M * \partial_M d^2$. The derivative $\partial_M d^2$ is calculated as follows:

$$\begin{aligned} \partial_M d^2 &= -\frac{1}{2} \partial_M \left\langle M\Lambda\widetilde{M} p M\Lambda\widetilde{M} p \right\rangle \\ &= -\left\langle \Lambda\widetilde{M} p M\Lambda\widetilde{M} p \right\rangle_M + \left\langle \widetilde{M} p M\Lambda\widetilde{M} p M\Lambda\widetilde{M} \right\rangle_M \end{aligned} \tag{3}$$

where M must be a motor so $\widetilde{M} = M^{-1}$. The operator $\langle \ldots \rangle_M$ denotes the projection of a general multivector onto the grades being present in multivector M. The optimisation returns the estimated parameters x of the motor $M(x)$.

Root finding of a point-line constraint equation. An alternative distance measure is expressed in an implicit way by the equation

$$p \wedge (M\Lambda\widetilde{M}) = 0 \tag{4}$$

which indicates point p is on line Λ. We call this the *point-line constraint.*

For all target points, the point-line constraint becomes

$$\sum_i \alpha_i \left(\sum_j p_i \wedge (M\Lambda_j\widetilde{M}) \right) = 0 \tag{5}$$

where $\alpha_i \in \{0, 1\}$ indicates whether the target is visible by any of the cameras. This point-line constraint expresses a geometric distance measure and is commonly applied in computer vision, see [5,8].

This technique uses the point-line constraint in Equation (4) for distance measurement. Given the previous motor M_{k-1}, M_k can be estimated as $M_k = \Delta M_k M_{k-1}$. Assume the previous pose M and line Λ are known. Let us update the current pose $\Delta M M$. The constraint becomes

$$\left(\widetilde{\Delta M} p' \Delta M\right) \wedge \Lambda = 0 \tag{6}$$

where $p' = \widetilde{M} p M$ represents a point in the previous CSM. ΔM needs to be estimated.

In order to solve for ΔM, it is necessary to linearise the motor part (i.e., $\widetilde{\Delta M} p' \Delta M$) of the equation. The motion of the camera group is considered as a general motion, which is formulated using an exponentiated bivector (2-vector); ΔM is expressed in the form

$$\exp\left(-\frac{\Delta B - \Delta t e}{2}\right)$$

where ΔB is a Euclidean bivector and Δt is a vector.

The Euclidean transformation (i.e., ΔM) of a point p' can be approximated as follows:

$$\begin{aligned} \widetilde{\Delta M} p' \Delta M &= \exp\left(\frac{\Delta B - \Delta t e}{2}\right) p' \exp\left(-\frac{\Delta B - \Delta t e}{2}\right) \\ &\approx \left(1 + \frac{\Delta B - \Delta t e}{2}\right) p' \left(1 - \frac{\Delta B - \Delta t e}{2}\right) \\ &\approx p' - p' \rfloor \Delta B + p' \rfloor (\Delta t e) \end{aligned} \tag{7}$$

In Equation (7), two approximations are involved. The first approximation involves truncating the Taylor series for $\exp(X)$ (i.e., $\exp(X) \approx 1 + X + \frac{X^2}{2!} + \cdots$). The second approximation involves removing second order terms from the final product; this works well only when the motion ΔM is sufficiently small (say, its rotation angle is smaller than 10 degree). This condition is satisfied when the camera group moves "smoothly" along its path and is sampled sufficiently frequently.

A similar linearisation of a transformation for a single point using different expressions is described in [8]. By substituting the approximated expression of $\widetilde{\Delta M}p'\Delta M$ given by Equation (7) back into constraint Equation (6), the constraint becomes

$$p' \wedge \Lambda - (p' \rfloor \Delta B) \wedge \Lambda + (p' \rfloor (\Delta t e)) \wedge \Lambda = 0 \tag{8}$$

with two unknowns: ΔB and Δt. Therefore, ΔM is calculated by estimating ΔB and Δt. A set of point-line correspondences are required to solve for ΔB and Δt in Equation (8). As any linear geometric algebra equation can be expressed in matrix form, we solve the equation by solving the associated matrix system of the form $Ax = b$. This can be solved by any standard technique such as LU decomposition. From x we obtain ΔB and Δt and hence ΔM. Each calculated ΔM provides a step towards the desired motor and this process is repeated until convergence. The first step towards the target motor is denoted by ΔM_1. By repeating this procedure, $M_{k2}, \ldots, M_{kn}$ are estimated, which converge towards M_k where n iterations are necessary. ΔM is calculated as $\Delta M_n \ldots \Delta M_2 \Delta M_1$. The convergence rate depends on the "speed" of the expected transformation (i.e., the movement of the cameras within the space where images are taken). We stop the approximation (iteration) if $\|\Delta M_i\| \leq \epsilon$ (e.g., $\epsilon = 10^{-6}$), which indicates that no further improvement can be achieved. Several iterations are usually sufficient to obtain the next pose of the camera group.

2.2 Update Target Positions

With the estimated pose M of CSM, the given lines Λ in CSM can be transformed to CSW by $M\Lambda\widetilde{M}$. Given all the lines in CSW for all poses, the current target positions can be calculated by Lemma 1 [9],

Lemma 1. *Let $\Lambda_j \in \mathcal{G}^3_{4,1}$, $j \in J$ be a set of normalised lines and $S(x) = \sum_{j \in J} S(x, \Lambda_j)$ where $S(x, \Lambda_j) = x - (x \rfloor \Lambda_j) \rfloor \Lambda_j$. If $\underline{S} I_3 \neq 0$ then the point $q \in \mathcal{G}^1_{4,1}$ closest to all the lines in the least squares sense is given by the center of the normalised dual sphere*

$$s = -\frac{\underline{S}(I_3) \rfloor I_4}{\underline{S}(I_3) \rfloor I_3} \tag{9}$$

where $I_3 = e_1 \wedge e_2 \wedge e_3$ and $I_4 = e_o \wedge e_1 \wedge e_2 \wedge e_3$.

As the target positions are estimated in real time, an increasingly large number of lines and frequently repeated calculations would require too much computational resource. Rather than storing all the lines we update some summary variables to implement an iterative algorithm.

In Lemma 1, $\underline{S}(I_3)$ and $\underline{S}(I_4)$ depend on all lines and vary with each update. As $\underline{S}(I_3) = S(e_1) \wedge S(e_2) \wedge S(e_3)$ and $\underline{S}(I_4) = S(e_o) \wedge S(e_1) \wedge S(e_2) \wedge S(e_3)$ it is only necessary to store and update $S(e_o)$, $S(e_1)$, $S(e_2)$ and $S(e_3)$. During the iterations, the information contained in the lines needed for estimating the target positions, are accumulated in $S(e_o)$, $S(e_1)$, $S(e_2)$ and $S(e_3)$. Recall S is defined as $S(q) = \sum_{i=1}^{n}(q - (q\rfloor\Lambda_i)\rfloor\Lambda_i)$. The current estimate of $S(e_j)$ can be updated based on previous $S_{k-1}(e_j)$, and new lines $\Lambda_i, i \in I_k$ arriving at current time k as

$$S_k(e_j) = S_{k-1}(e_j) + \sum_{i\in I_k}(e_j - (e_j\rfloor\Lambda_i)\rfloor\Lambda_i) \tag{10}$$

It is not necessary to update the targets on every pose update iteration. For example, the targets may be updated after CSM has been moved by some specified distance.

3 Experiments

Experiments were carried out using both simulated data and real data. Both kinds of data allowed us to test the validity and performance of our algorithm using both the point-line constraint and the objective function (Quasi-Newton optimisation) pose update. Noise was added to test the stability of the algorithm.

3.1 Simulated Data

In order to test and evaluate the iterative algorithm for estimating target positions, we generated simulated line data. We have the ground truth target position obtained using a total station. We generated a synthetic path for CSM in a real scene (a lab at Industrial Research Ltd.). Synthetic lines were created using this path and projecting the known targets through the real calibrated camera group model. In order to test the behaviour of the algorithm in the presence of noise we generated simulated data with different levels of noise. The stability of the algorithm is investigated by adding Gaussian noise with deviation $\sigma \in [0.2, 1.0]$ pixels (see Figure 2).

With the minimum noise, the errors of estimation decrease smoothly by around 30%. With more noise, the error curve fluctuates within a wider range. But the error is still reduced as the update process continues. Even with the maximum noise, the target position is refined by around 20%. We applied the simulated data to both algorithms. Both algorithms are validated by a comparison of experimental results with ground truth, and also between both. Table 1 shows comparisons for estimating different poses of CSM along a 3D path.

Comparisons showed that both pose update strategies achieve almost the same results. The strategy using the point-line constraint was nearly twice as fast as Quasi-Newton strategy. This can be partly attributed to the fact that the point-line constraint method make no effort to guarantee global convergence. The Quasi-Newton method proved more robust under all considered conditions, and the point-line constraint method is limited to the condition that differences between subsequent poses are small because no global convergence protection was implemented.

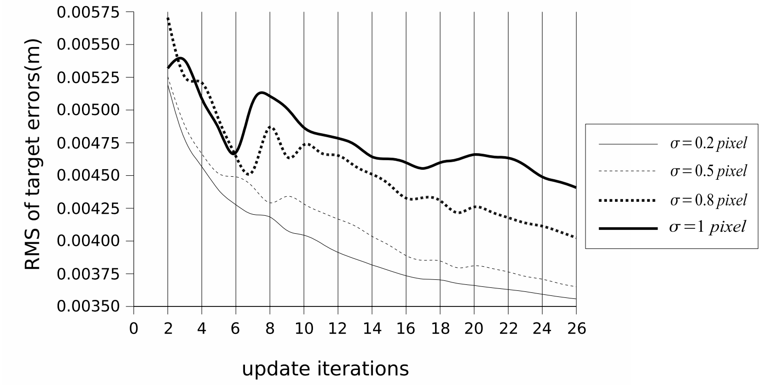


Fig. 2. The *RMS* (Root Mean Square) of errors in targets vs update iterations with different levels of noise

Table 1. Comparison of results for the two alternative pose estimation strategies for the kth pose (rotation and translation). θ_1 and t_1 are rotation angle (in degree) and translation vector (in millimeter) of the pose using the quasi-Newton method; θ_2 and t_2 are those for the line-target constraint method.

k	$(\theta_1 - \theta_2) \times 10^{-4}$	$\lvert t_1 - t_2\rvert \times 10^{-4}$
1	1.23	1.44
5	2.11	0.84
10	0.67	2.10
15	1.01	1.25
20	0.19	0.09
26	3.61	0.56

3.2 Real Data

Real data sequences of images captured by the camera group are shown in Figure 1, right. The lab room is visualised using VRML software; see Figure 3. Results for real data were not as good (for both pose update strategies) as for simulated data.

We believe that this can be partially explained by small errors in the camera group model. A better camera group calibration should reduce these errors. During simulation the same camera group model is used for projection (targets mapped to image points) and backprojection (image points mapped to lines) so any calibration errors have no influence.

Estimated poses and target positions are also visualised in Figure 3. Comments about performance comparisons between both estimation methods apply qualitatively for real data the same way as for simulated data. The target update algorithm run run at 30Hz on a standard 3GHz PC using either of the pose update schemes.

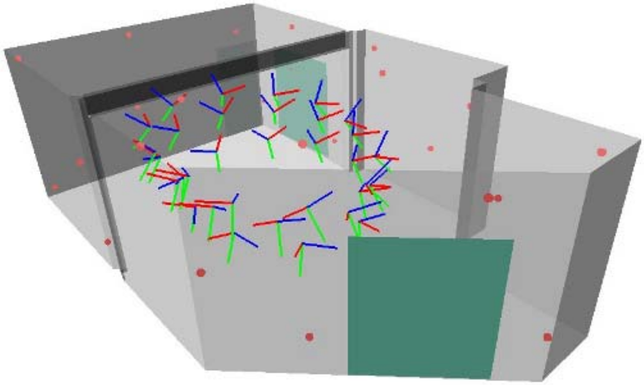


Fig. 3. A model of the lab space used. Disks are estimated targets; the figure also shows a few *CSM* coordinate systems along the path of the camera group.

4 Conclusion

We developed an iterative algorithm for refining 3D target positions over a large number of images. We acquire (from 2D images) lines pointing towards 3D targets. The use of the conformal model of geometric algebra (CGA) benefits the development of the solution in both theory and practice. CGA provides a compact symbolic representation of objects and their transformations. A variety of objects (e.g., vectors, points, lines, spheres) and operations (e.g. motors) can be represented in a single algebra which simplifies the implementation. The use of a single motor element to represent a Euclidean transformation (instead of separate rotation and translation), further simplified the implementation.

The iterative target update algorithm performed well over a wide variety of conditions. Two iterative strategies are used for pose estimation. The point-line constraint strategy proved to be more efficient than the Quasi-Newton optimisation strategy, but less robust in stability.

References

1. L. Dorst and S. Mann. Geometric algebra: a computational framework for geometrical applications, Part 1 and Part 2. *IEEE Computer Graphics Applications*, vol. 22, no. 3 and 4, 2002.
2. D. Hestenes, Old wine in new bottles: A new algebraic framework for computational geometry, In E. Bayro-Corrochano and G. Sobczyk, editors, *Geometric Algebra with Applications in Science and Engineering*, chapter 1, Birkhäuser, 2001.
3. T.F. Havel, Geometric algebra: parallel processing for the mind, In: *MIT Independent Activities Period Lectures*, 2002.
4. W. E. L. Grimson. *Object Recognition by Computer.* The MIT Press, Cambridge, MA, 1990.
5. G. A. Kramer. *Solving Geometric Constraint Systems: A Case Study in Kinematics.* ACM Distinguished Dissertations, MIT Press, 1992.

6. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++*, 2nd Edition, Cambridge University Press, 2002.
7. B. Rosenhahn and G. Sommer. Pose estimation in conformal geometric algebra. Part I: The stratification of mathematical spaces. *J. Mathematical Imaging Vision*, **22**:27–48, 2005.
8. B. Rosenhahn and G. Sommer. Pose estimation in conformal geometric algebra. Part II: Real-time pose estimation using extended feature concepts. *J. Mathematical Imaging Vision*, **22**:49–70, 2005.
9. R. J. Valkenburg and N. S. Alwesh. Calibration of target positions using the conformal model and geometric algebra. In Proc. *Image Vision Computing New Zealand*, Otago University, pages 241-246, 2005.
10. R. J. Valkenburg, Some techniques in geometric algebra for computer vision, Tech. Rep. 87130001-1-03, Industrial Research Limited, August, 2003.
11. R. Wareham, J. Cameron, and J. Lasenby. Applications of conformal geometric algebra in computer vision and graphics. In Proc. *IWMM/GIAE*, pages 329–349, 2004.

Robust Pose Estimation with 3D Textured Models

Juergen Gall, Bodo Rosenhahn, and Hans-Peter Seidel

Max-Planck Institute for Computer Science
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
{jgall, rosenhahn, hpseidel}@mpi-sb.mpg.de

Abstract. Estimating the pose of a rigid body means to determine the rigid body motion in the 3D space from 2D images. For this purpose, it is reasonable to make use of existing knowledge of the object. Our approach exploits the 3D shape and the texture of the tracked object in form of a 3D textured model to establish 3D-2D correspondences for pose estimation. While the surface of the 3D free-form model is matched to the contour extracted by segmentation, additional reliable correspondences are obtained by matching local descriptors of interest points between the textured model and the images. The fusion of these complementary features provides a robust pose estimation. Moreover, the initial pose is automatically detected and the pose is predicted for each frame. Using the predicted pose as shape prior makes the contour extraction less sensitive. The performance of our method is demonstrated by stereo tracking experiments.

1 Introduction

This paper addresses the task of estimating the pose of a rigid body in the 3D space from images captured by multiple calibrated cameras. For solving this problem it is a natural approach to exploit the available information on the object as far as possible. In [1] the knowledge of the 3D shape was integrated in a contour based 3D tracker. Knowing the 3D model, the estimating process relies on correspondences between some 2D features in the images and their counterparts on the 3D model. Our approach extends the work by incorporating also the texture of the object. The additional information allows to extract more reliable correspondences that makes the estimation more robust.

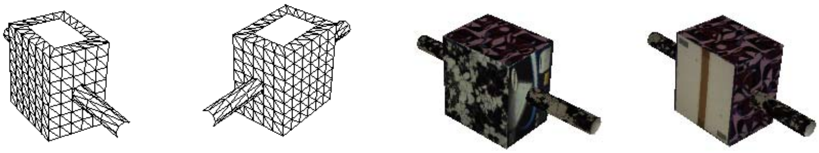


Fig. 1. 3D mesh and rendered textured model used for tracking

There are numerous features that have been used for establishing correspondences, e.g., matching lines [2], blocks [3], local descriptors [4], and free-form contours [5]. They all work well under some conditions, however, none of them can handle general situations. The most approaches assume that the corresponding image features are visible during the whole sequence. They either completely fail when the number of features

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 84–95, 2006.

is very low caused, for example, by occlusion or they reinitialize the pose after some frames when enough features are again detected [6]. Whereas the contour extraction as described in [1] is robust to occlusions. However, the contour does not provide enough information for smooth and convex objects to estimate the pose uniquely. Furthermore, the contour extraction is only suitable for movements that are slow enough such that the segmentation does not get stuck in a local optimum. Hence, more than one feature is needed for robust tracking.

Combining the object contour with the optical flow between successive frames has been proposed in [7]. Although it performs well, it assumes that the initial pose is known and cannot recover from a significant error. Furthermore, the optical flow is easily distracted by other objects moving in front of the observed object. Our work instead combines the object contour with image features between a frame and a 3D textured model projected onto the image plane. We assume that the textured model of the object is available where the lightning conditions for capturing the texture are allowed to differ from the conditions during tracking, i.e., the model construction is independent of the tracking sequence.

Since lightning conditions between the object and its textured model are inhomogeneous and the object is transformed by a rigid body motion (RBM), we use local descriptors that provide robust matching under changes in viewpoint and illumination. A comparison of local descriptors [8] revealed that SIFT [9], PCA-SIFT [10], and GLOH [8] perform best. The descriptors build a distinctive representation of a so-called keypoint in an image from a patch of pixels in its neighborhood. The keypoints are localized by an interest point detector. We use the detector proposed by Lowe [11] based on local 3D extrema in the scale-space pyramid built with difference-of-Gaussian filters. It has the advantage that it runs faster than other detectors [12], e.g., like the slower Harris-Affine detector [13]. The DoG representation, however, is not affine invariant. Hence, we cannot use GLOH that requires an affine-invariant detector. Therefore, we used PCA-SIFT that reduces the dimension of the descriptor by principal component analysis. This speeds up the matching process and produces less outliers than SIFT but also less correspondences.

In the next section, we give an overview of the whole pose estimation process that will be discussed in detail in the following sections. Experiments in Section 5 with a 3D textured model as shown in Fig. 1 demonstrate the performance of the proposed technique. A brief discussion is given at the end.

2 Overview

Our approach for pose estimation is illustrated by the flow chart in Fig. 2. Knowing the pose of the object for frame $t - 1$, we generate a 3D textured model in the same world coordinate system used for the calibration of the cameras, see Section 4.1. Rendered images of the model are obtained by projecting the model onto the image plane according to the calibration matrix for each camera.

In a second step, the PCA-SIFT [10] features are extracted from the rendered images and from the new images of frame t. The features are used for establishing correspondences between the 3D model and the 2D images for each view as described in

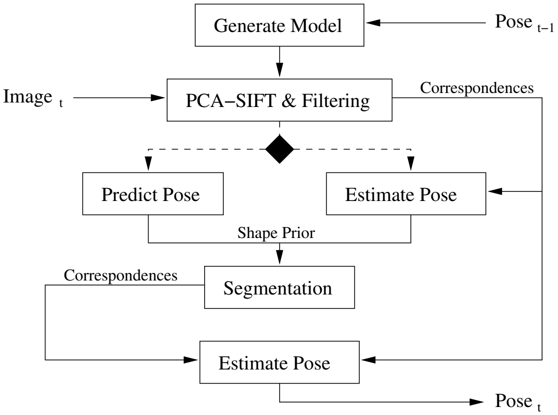


Fig. 2. Correspondences extracted by PCA-SIFT and correspondences between the contour of the projected 3D model and the contour obtained by segmentation are used for pose estimation. If not enough keypoints are detected by PCA-SIFT, an autoregression is performed to predict the pose for the next frame.

Section 4.2. In [6] and [14], RANSAC is used to estimate the pose from the matches that include outliers. RANSAC, however, is not suitable for integrating correspondences from the contour and cannot handle inaccuracy of the keypoint localizations, e.g., arising from texture registration. Therefore, we use a least-squares approach as used in [5], see Section 3. If not enough correspondences are extracted by PCA-SIFT, the pose is predicted by autoregression as discussed in Section 4.3.

The next step consists of extracting the contour by a variational model for level set based image segmentation incorporating color and texture [15] where the predicted pose is used as shape prior [1], see Section 4.4. New correspondences between the 3D model and the 2D image are then established by matching the extracted contour with the projected contour of the model via an iterated closest point algorithm [16]. Finally, the correspondences obtained from PCA-SIFT and from the segmentation are used for estimating the pose in frame t.

3 Pose Estimation

For pose estimation we assume that correspondences between the 3D model (X_i) and a 2D image (x_i) are already extracted and write each correspondence as pair (X_i, x_i) of homogeneous coordinates. In order to estimate the 3D rigid body motion M that fits best the correspondences, M is represented as exponential of a twist [17]

$$\theta\hat{\xi} = \theta \begin{pmatrix} \hat{\omega} & v \\ 0 & 0 \end{pmatrix}, \qquad \hat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \qquad \|\omega\|_2 = 1, \tag{1}$$

i.e., $M = \exp(\theta\hat{\xi})$. A twist with varying $\theta \in \mathbb{R}$ describes a screw motion in $\mathbb{R}^3$ where θ corresponds to the rotation velocity and pitch. The function $\exp(\theta\hat{\xi})$ can be efficiently computed by the Rodriguez formula [17] and linearized by $\exp(\theta\hat{\xi}) = \sum_{k=0}^{\infty}((\theta\hat{\xi})^k/k!)$ $\approx I + \hat{\xi}$, where I denotes the identity matrix.

Each image point x_i defines a projection ray that can be represented as Plücker line [17] determined by a unique vector n_i and a moment m_i such that $x \times n_i - m_i = 0$ for all x on the 3D line. Furthermore, $\|x \times n_i - m_i\|_2$ is the norm of the perpendicular error vector between the line and a point $x \in \mathbb{R}^3$. Hence, the pose estimation consists of finding a twist such that the squared error for $(\exp(\theta\hat{\xi})X_i)_{3\times1}$ is minimal for all pairs, where $(\cdot)_{3\times1}$ denotes the transformation from homogeneous coordinates back to non-homogeneous coordinates. Using the linearization, we obtain for each correspondence the constraint equation

$$(\exp(\theta\hat{\xi})X_i)_{3\times1} \times n_i - m_i = 0 \qquad (2)$$

which can be rearranged into the form $A\xi = b$. The resulting overdetermined linear system is solved by standard methods like the Householder algorithm. From the resulting twist ξ, the RBM M_1 is computed and applied to all X_i. The pose estimation is iterated until the motion converges. After n iterations, usually 3-5 are sufficient, the concatenated rigid body transformation $M = M_n \ldots M_2M_1$ is the solution for the pose estimation. In a multi-view setting as in our experiments, the correspondences for each camera are added to one linear system and solved simultaneously. Our implementation takes about 4ms for 200 correspondences.

4 Correspondences

4.1 Textured Model

We assume that a 3D model including textures is already constructed independently of the tracking sequences, i.e., we do not require that the textures are extracted from the tracking sequences. Hence, the modelling process is done only once and the model can be reused for any sequence provided that the texture remains unchanged. In order to render the 3D model in the same coordinate system as used for camera calibration, the calibration matrices are converted to the modelview and projection matrix representation of OpenGL. Since OpenGL cannot handle lens distortions directly, the image sequences are undistorted beforehand. However, the step could also be efficiently included by a look-up table. In a preprocessing step, PCA-SIFT is trained for the object by building the patch eigenspace from the object textures. Moreover, we render some initial views of the 3D model by rotating and store the extracted keypoints, strictly speaking the PCA-SIFT descriptors of the keypoints, with the corresponding RBM. From the data, our system automatically detects the pose in the first frame.

4.2 Matching

After the 3D model is rendered and projected onto the image plane for each camera view, the keypoints are extracted by PCA-SIFT. The keypoints are also extracted from the captured images. The effort is reduced by bounding cubes for each component of the 3D model. Projecting the corners of the cubes provides a 2D bounding box for each

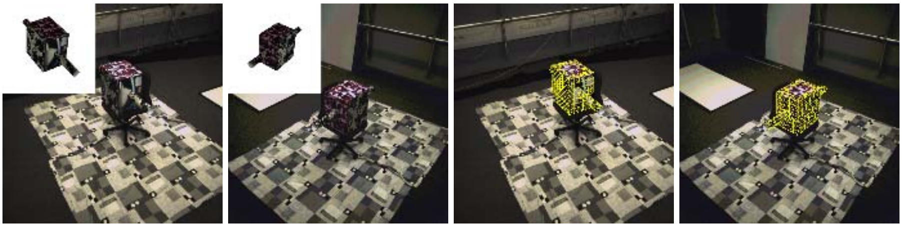


Fig. 3. Initialization. **Left:** Both camera views of the first frame. Best initial view for initialization is shown in top left corner. **Right:** Estimated pose after initialization.

image. Since we track an object, we can assume that the object is near the bounding box except for the first frame. Hence, the detector is only performed on a subimage. 2D-2D correspondences are then established by nearest neighbor distance ratio matching [8], where we use as additional constraint that two different located points cannot correspond to points with the same position. Since the set of correspondences contains outliers, the rudest mismatches are removed by discarding correspondences with an Euclidean distance that exceeds the average by a multiple.

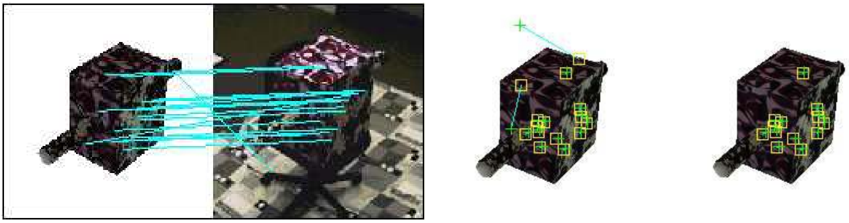


Fig. 4. Left: Correspondences between projected model and image. **Center:** Displaying the points of the projected model *(yellow squares)* corresponding to points in the image *(green crosses)*. Two outliers are in the set of correspondences. **Right:** After filtering only the outliers are removed.

The 3D coordinate X of a 2D point x in the projected image plane of the model is obtained as following: Each 2D point is inside or on the border of a projected triangle of the 3D mesh with vertices v_1, v_2, and v_3. The point can be expressed by barycentric coordinates, i.e., $x = \sum_i \alpha_i \, v_i$. Assuming an affine transformation, the 3D point is given by $X = \sum_i \alpha_i \, V_i$. The corresponding triangle for a point can be efficiently determined by a look-up table containing the color index and vertices for each triangle. After that the pose is estimated from the resulting 2D-3D correspondences. In a second filtering process, the new 3D coordinates from the estimated pose are projected back and the last outliers are removed by thresholding according to the Euclidean distance between the 2D correspondences and the reprojected counterparts.

During initialization, the keypoints from the images are matched with the keypoints extracted from the initial views beforehand. According to the number of matches, a best initial view is selected and the pose is estimated from the obtained correspondences.

4.3 Prediction

The logarithm of a RBM: In [17] a constructive way is given to compute the twist which generates a given RBM: Let $R \in SO(3)$ be a rotation matrix and $t \in \mathbb{R}^3$ a translation vector for the RBM. For the case $R = I$, the twist is given by

$$\hat{\xi} = \begin{pmatrix} 0 & \frac{t}{\|t\|} \\ 0 & 0 \end{pmatrix}, \theta = \|t\|_2. \tag{3}$$

For the other cases, the motion velocity θ and the rotation axis ω is given by

$$\theta = \cos^{-1}\left(\frac{trace(R) - 1}{2}\right), \omega = \frac{1}{2\sin(\theta)}\begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}. \tag{4}$$

To obtain v, the matrix

$$A = (I - \exp(\theta\hat{\omega}))\hat{\omega} + \omega\omega^T\theta, \tag{5}$$

obtained from the Rodriguez formula needs to be inverted and multiplied with the translation vector t, i.e., $v = A^{-1}t$. This follows from the fact, that the two matrices which comprise A have mutually orthogonal null spaces when $\theta \neq 0$. Hence, $Av = 0 \Leftrightarrow v = 0$. We call the transformation from $SE(3)$ to $se(3)$ the logarithm, $\log(M)$.

The adjoint transformation: It is not trivial to derive a formula for the velocity of a rigid body whose motion is given by $g(t)$, a curve parameterized by time t in $SE(3)$, since $SE(3)$ is not Euclidean. In particular, $\dot{g} \notin SE(3)$ and $\dot{g} \notin se(3)$. But by representing a rigid body motion as a screw action, the spatial velocity can be represented by the twist of the screw, see [17] for details. This allows for motion interpolation, damping and prediction.

Later we will take the motion history P_i of the last N frames into account. For a suited prediction we use a set of twists $\xi_i = \log(P_i P_{i-1}^{-1})$ representing the relative motions. To generate a suited *average* rigid body motion we make use of the adjoint transformation to represent a screw motion with respect to another coordinate system: If $\xi \in se(3)$ is a twist given in a coordinate frame A, then for any $G \in SE(3)$ which transforms a coordinate frame A to B, is $G\hat{\xi}G^{-1}$ a twist with the twist coordinates given in the coordinate frame B, see [17] for details. The mapping $\hat{\xi} \longmapsto G\hat{\xi}G^{-1}$ is called the *adjoint transformation* associated with G.

Given a set of world positions and orientations P_i the twists ξ_i can be used to express the motion as local transformation in the current coordinate system M_1: Let $\xi_1 = \log(P_2 P_1^{-1})$ be the twist representing the relative motion from P_1 to P_2. This transformation can be expressed as local transformation in the current coordinate system M_1 by the adjoint transformation associated with $G = M_1 P_1^{-1}$. The new twist is then given by $\hat{\xi}_1' = G\hat{\xi}_1 G^{-1}$. The advantage of the twist representation is now that the twists can be scaled by a factor $0 \leq \lambda_i \leq 1$ to damp the local rigid body motion, i.e., $\hat{\xi}_1' = G\lambda_1\hat{\xi}_1 G^{-1}$.

The average RBM from N given local rigid body motions can then be written as consecutive evaluation of such local rigid body motions scaled with $\lambda_i = 1/N$.

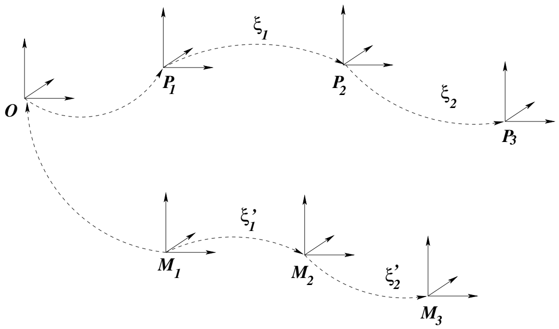


Fig. 5. Transformation of rigid body motions from prior data P_i in a current world coordinate system M_i. A proper scaling of the twists results in a proper damping.

4.4 Segmentation

The images are segmented by a level set based method incorporating color and texture [15]. It splits the image domain Ω^i of each view into object region Ω_1^i and background region Ω_2^i by level set functions $\Phi^i : \Omega^i \rightarrow \mathbb{R}$, such that $\Phi^i(x) > 0$ if $x \in \Omega_1^i$ and $\Phi^i(x) < 0$ if $x \in \Omega_2^i$. The contour of an object is thus represented by the zero-level line. The approach described in [1] uses a variational model that integrates the contour of a prior pose $\Phi_0^i(\widehat{x})$ for each view $1 \leq i \leq r$ as shape prior. It minimizes the energy functional $E(\widehat{x}, \Phi^1, \ldots, \Phi^r) = \sum_{i=1}^r E(\widehat{x}, \Phi^i)$ where

$$\begin{aligned} E(\widehat{x}, \Phi^i) = & - \int_{\Omega^i} H(\Phi^i) \ln p_1^i + (1 - H(\Phi^i)) \ln p_2^i \, dx \\ & + \nu \int_{\Omega^i} \left| \nabla H(\Phi^i) \right| dx + \lambda \int_{\Omega^i} \left(\Phi^i - \Phi_0^i(\widehat{x}) \right)^2 dx \end{aligned} \tag{6}$$

and H is a regularized version of the step function.

Minimizing the first term corresponds to maximizing the a-posteriori probability of all pixel assignments given the probability densities p_1^i and p_2^i of Ω_1^i and Ω_2^i, respectively. These densities are modeled by Gaussian densities whose parameters are estimated from the previous level set function. The second term minimizes the length of the contour and smoothes the resulting contour. The last one penalizes the discrepancy to the shape prior that is obtained by projection of the predicted pose. The relative influence of the three terms is controlled by the constant weighting parameters $\nu = 0.5$ and $\lambda = 0.06$.

After segmentation, the 3D-2D correspondences for each view are given by the projected vertices of the 3D mesh that are part of the model contour and their closest points of the extracted contour determined by an iterated closest point algorithm [16].

4.5 Fusion of Correspondences

Although it has been shown that the segmentation as previously described is quite robust to clutter, shadows, reflections, and noise [1], a good shape prior is essential for tracking

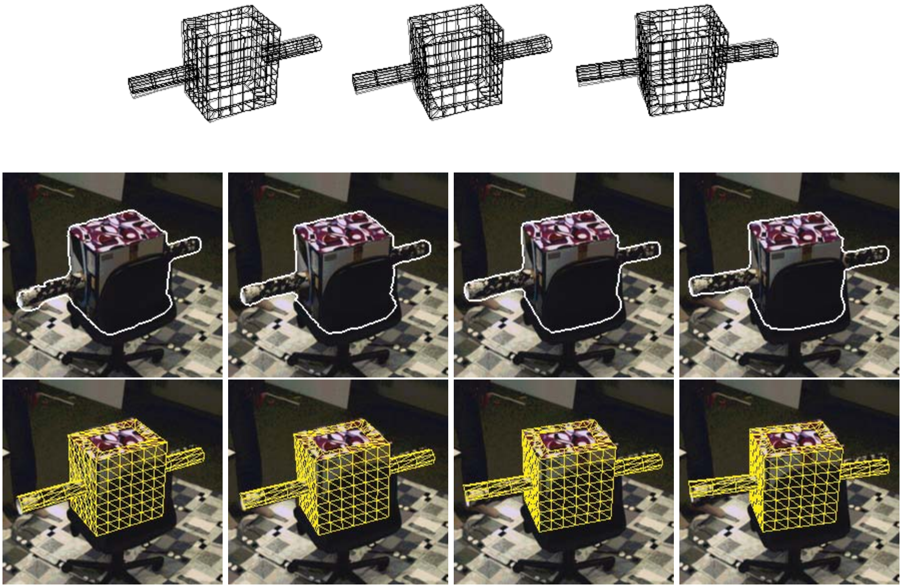


Fig. 6. 4 successive frames of a rotation sequence (only one view is shown). **Top row:** Pose is predicted by autoregression for lack of PCA-SIFT matches. *Black:* Predicted pose. *Gray:* Previous pose. **Middle row:** Contour extracted by segmentation. **Bottom row:** Estimated pose.

since both matching between the contours and the segmentation itself is prone to local optima. The predicted pose by an autoregression usually provides a better shape prior than the estimated pose in the previous frame. In situations, however, where the object region and the background region are difficult to distinguish, the error of the segmentation and the error of the prediction are accumulating after some time. The shortcoming

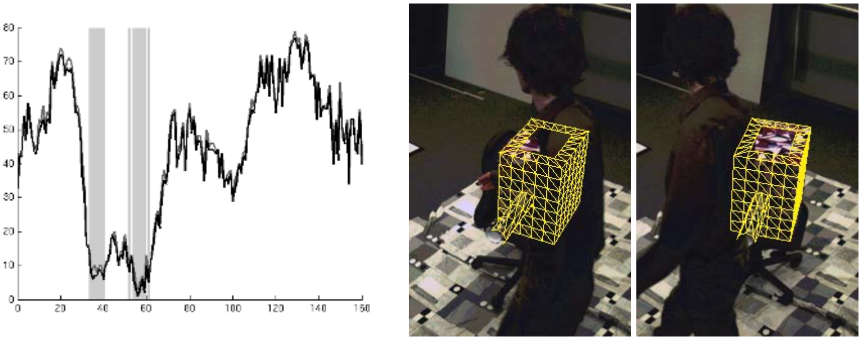


Fig. 7. Rotation sequence with a moving person. **Left:** Number of matches from PCA-SIFT *(dark gray)*. After filtering the number of matches is only slightly reduced *(black)*. When the number is below a threshold, the pose is predicted by an autoregression *(gray bars)*. **Right:** The rotating box is occluded by a moving person.

is compensated by PCA-SIFT, but it is also clear that usually not enough keypoints are available in each frame. Hence, the correspondences from contour matching and from descriptor matching are added to one linear system for the pose estimation. Since the contour provides more correspondences, the Equations (2) for the correspondences from PCA-SIFT are weighted by $\sharp Corrs_{Contour}/5$.

5 Experiments

For evaluating the performance of our approach, we used the 3D textured model as shown in Fig. 1. The textures were captured under different lightning conditions from the conditions for the image sequences that were recorded by two calibrated cameras. Although the size of the images is 502×502, the object is only about 100×100. The initial position was automatically detected for each sequence as shown in Fig. 3.

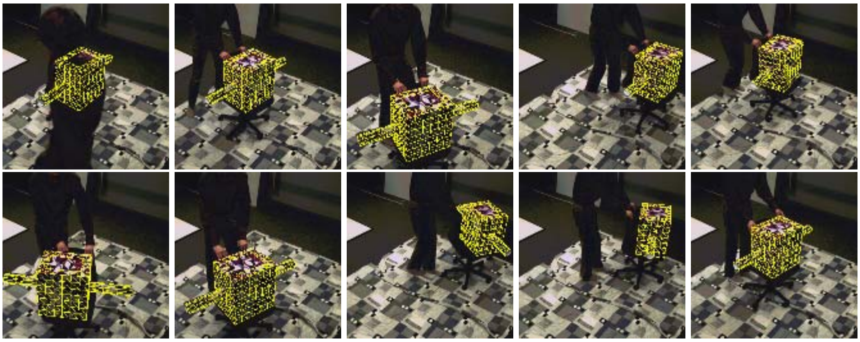


Fig. 8. Pose estimates for 10 of 570 frames. The sequence contains several difficulties for tracking: a rich textured and non-static background, shadows, occlusions, and other moving objects. Only one camera view is shown.

The tracked object is partially covered with two dissimilar customary fabrics and the printed side reflects the light. It is placed on a chair that occludes the back of the object. The background is rich textured and non-static. Shadows, dark patterns on the texture and the black chair make contour extraction difficult even for the human eye. Furthermore, a person moves and occludes the object. These conditions make great demands on the method for pose estimation.

In the first sequence, the chair with the object rotates clockwise. When the back of the chair occludes the object, there are not enough distinctive interest points for pose estimation. Therefore, the pose is predicted by an autoregression for the next frame as shown in Fig. 6. Due to the shape prior, the segmentation is robust to the occlusion such that the estimates are still accurate. The number of matches from PCA-SIFT with respect to time is plotted in Fig. 7. During the sequence, the object rotates counterclockwise while the person orbits the object clockwise. As we can see from the diagram, PCA-SIFT produces only few outliers that are removed after the filtering. The gray bars in the diagram indicate the frames where an autoregression was performed. Since the

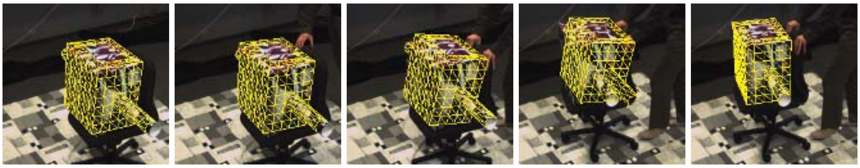


Fig. 9. Comparison with a contour-based method. **From left to right:** Pose estimates for frames 5, 50, 90, 110. **Rightmost:** Result of our method at frame 110.

number of matches range from 1 to 77, it is clear that an approach based only on the descriptors would fail in this situation.

Pose estimates for a third sequence including rotations and translations of the object are shown in Fig. 8. When only the contour is used, the pose estimation is erroneous since both segmentation and contour matching are distracted by local optima, see Fig. 9. For comparison, the result of our method is also given.

Finally, we simulated disturbances of the sequence in order to obtain a quantative error analysis. Since the object is placed on the chair, the y-coordinate of the pose is approximately constant. During the sequence, however, the object shifts slightly on the chair. The peak at frame 527 in the diagram of Fig. 10 is caused by a relocation of the object. For one sequence, we added Gaussian noise with standard deviation 35 to each color channel of a pixel. Another sequence was disturbed by 80 teapots that were rendered in the 3D space of the tracked object. The teapots drop from the sky where the start positions, material properties, and velocities are random. Regarding the result for the undistorted sequence as some kind of ground truth, the diagram in Fig. 10 shows the robustness of our approach. While an autoregression was performed only twice for the unmodified sequence and the average number of filtered matches per frame from PCA-SIFT was 50.9, the numbers fell down to 27.9 and 13.1 for the teapots sequence with 132 predictions and the noisy sequence with 361 predictions.

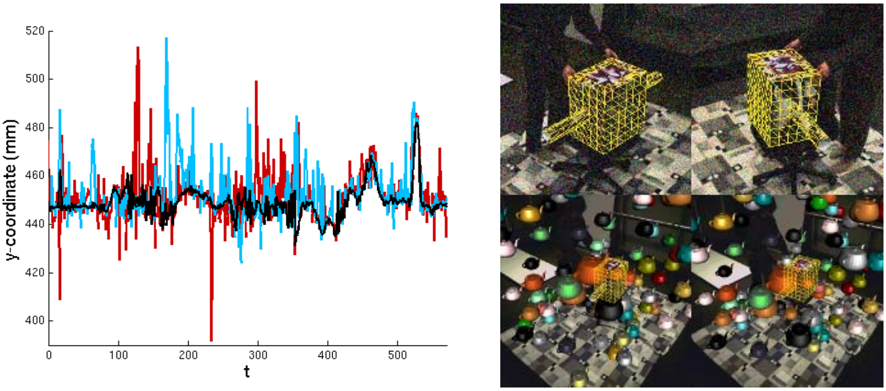


Fig. 10. Left: Quantative error analysis for a sequence with disturbances. *Black:* Undisturbed sequence. *Red:* Gaussian noise with standard deviation 35. *Blue:* 80 teapots dropping from the sky with random start position, material properties, and velocity. **Right:** *Top:* Stereo frame 527 of the noisy sequence (image details). *Bottom:* Two successive frames of the teapot sequence.

6 Conclusions

In this work, we have suggested a textured model based method for 3D pose estimation. It fuses two different features for matching, namely contour and local descriptors, where the influence of the features is automatically adapted during tracking. The initial pose is identified without supervision. In our experiments, we have demonstrated that our approach overcomes the drawbacks of the single features and that it can be applied to quite general situations. In the case of a homogeneous object without distinctive keypoints, our approach operates as a pure contour-based method. Furthermore, we have provided visual and quantative results showing that our approach is able to deal with a rich textured and non-static background and multiple moving objects. Moreover, it is robust to shadows, occlusions, and noise. Although our experiments considered only rigid bodies with a simple geometric surface, our method works with any kind of free-form objects. The pose estimation can be straightforward extended to articulated objects [18]. This will be done in future.

Acknowledgments

Our research is funded by the Max-Planck Center for visual computing and communication. We thank Yan Ke for providing the PCA-SIFT source code.

References

1. Rosenhahn, B., Brox, T., Weickert, J.: Three-dimensional shape knowledge for joint image segmentation and pose tracking. Int. J. of Computer Vision (2006)
2. David, P., DeMenthon, D., Duraiswami, R., Samet, H.: Simultaneous pose and correspondence determination using line feature. In: Int. Conf. of Computer Vision. (2003) 424–431
3. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3d tracking using online and offline information. IEEE Trans. on Pattern Analysis and Machine Intelligence **26**(10) (2004) 1391–1391
4. Allezard, N., Dhome, M., Jurie, F.: Recognition of 3d textured objects by mixing view-based and model-based representations. Int. Conf. on Pattern Recognition **01** (2000) 960–963
5. Rosenhahn, B., Perwass, C., Sommer, G.: Pose estimation of free-form contours. Int. J. of Computer Vision **62**(3) (2005) 267–289
6. Lepetit, V., Pilet, J., Fua, P.: Point matching as a classification problem for fast and robust object pose estimation. In: Conf. on Computer Vision and Pattern Recognition. Volume 2. (2004) 244–250
7. Brox T., Rosenhahn B., C.D., H.-P., S.: High accuracy optical flow serves 3-d pose tracking: Exploiting contour and flow based constraints. In Leonarids A., B.H., A., P., eds.: European Conf. on Computer Vision. Volume 3952 of LNCS., Springer (2006) 98–111
8. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Conf. on Computer Vision and Pattern Recognition **02** (2003) 257–263
9. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. of Computer Vision **60**(2) (2004) 91–110
10. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: IEEE Conf. on Computer Vision and Pattern Recognition. Volume 2. (2004) 506–513
11. Lowe, D.: Object recognition from local scale-invariant features. In: Int. Conf. on Computer Vision. (1999) 1150–1157

12. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. of Computer Vision **60**(1) (2004) 63–86
13. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: European Conf. on Computer Vision. (2002) 128–142
14. Brown, M., Lowe, D.: Invariant features from interest point groups. In: British Machine Vision Conf. (2002) 656–665
15. Brox, T., Rousson, M., Deriche, R., Weickert, J.: Unsupervised segmentation incorporating colour, texture, and motion. In Petkov, N., Westenberg, M.A., eds.: Computer Analysis of Images and Patterns. Volume 2756 of LNCS., Springer (2003) 353–360
16. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. Int. J. of Computer Vision (1994)
17. Murray, R., Li, Z., Sastry, S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton, FL (1994)
18. Rosenhahn, B., Brox, T., Smith, D., Gurney, J., Klette, R.: A system for marker-less human motion estimation. Künstliche Intelligenz **1** (2006) 45–51

Multi-scale 3D-Modeling

Karsten Scheibe[1], Michael Suppa[1], Heiko Hirschmüller[1],
Bernhard Strackenbrock[2], Fay Huang[3],
Rui Liu[1], and Gerd Hirzinger[1]

[1] German Aerospace Center, Oberpfaffenhofen and Berlin, Germany
[2] Illustrated Architecture, Berlin, Germany
[3] Yi-Lan University, Yi-Lan, Taiwan

Abstract. This paper reviews 3D-modeling activities at the German DLR Institute of Robotics and Mechatronics, carried out within the last decade in cooperation with partners in Germany (Z+F, Illustrated Architecture, DLR Institutes of Optical Information Systems, and of Planetary Research) and international partners. The main focus is on multisensory (e.g. push-broom or rotating stereo line cameras, laser range finders) information containing (at least) geometry and texture. The paper describes systems which acquire such information at different scales of scenery, ranging from indoor scenes to planetary explorations. It also covers principles and methods for preprocessing, geometric reconstruction, texture mapping, or matching.

1 Introduction

Photogrammetry (originally a method for recording and monitoring architecture) dates back to the work of Albrecht Meydenbauer. He was a German architect, who used a *graphic intersection method* for 3D analysis as early as in 1867. Photogrammetry is a measurement technology in which 3D coordinates of objects are determined by measurements, made in two or more photographic images taken at varying *attitudes* (i.e., position or viewing direction).

Modeling a 3D scene based on captured images, possibly including further sensors or interaction with the scene, can be achieved in different ways. The classical method in photogrammetry is (calibrated) *binocular stereo analysis.* Another, widely applied approach for 3D object or scene modeling is *structured lighting. Structure from motion* (SfM) is one of the more recent approaches within computer vision, which aims on estimating 3D structure from uncalibrated 2D image sequences.

A *laser rangefinder* (LRF) or *laser scanner* determines distances to opaque objects; it is also known as LIDAR (Laser Imaging Detection and Ranging). Such a device determines the distance to an object or surface using laser pulses (similar to radar technology, which uses radio waves instead of light). LRFs have been used for close-range photogrammetry (e.g., acquisition of building geometries) for several years, see [11].

Each of these approaches comes with particular limitations, and flexible solutions towards 3D scene recovery often apply multiple tools, do not restrict itself

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 96–107, 2006.

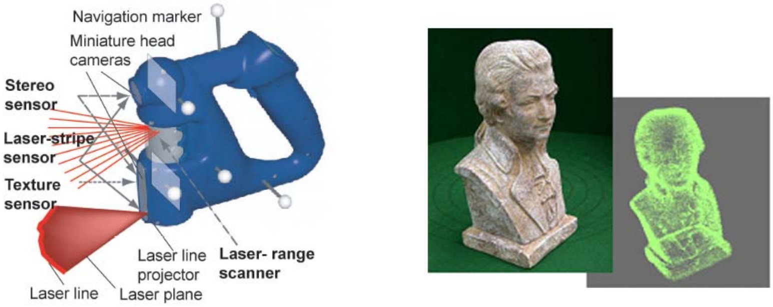


Fig. 1. DLR Multisensory 3D-Modeler and a scanned bust

to the use of cameras as the only sensor option, and merge different approaches as well; see, for example, [8]. 3D scene modeling based on using an LRF (active system) together with different digital cameras (texturing and stereo processing in different scales), and laser line projectors (structured light) are an example for multiple 3D recovering. Combining different sensors allows for acquiring any object with different levels of detail, e.g. fast digitization of a large object at medium resolution and refining some parts with higher accuracy afterwards. For example, in many robotics areas, depth information is used to avoid collisions, to navigate through an environment or to plan a grasp of an object. Texture information is used to auto-locate a robot in its environment as well as to find and identify objects. Various types of sensors and methods are needed for modeling different sized cultural objects. They range from small objects, like busts or statues, over medium scaled objects, like rooms and interiors of buildings, up to large terrains.

This paper gives an overview about recent (say, last decade) work at DLR Institutes (in cooperation with partners as stated in the abstract). Projects involve a large variety (by size, shape, or texture) of 3D objects [5].

2 3D Modeling Systems

This section deals with the first step when documenting 3D data about cultural heritage: the selection of the appropriate sensor technology. We categorize sceneries by sensing distances into three scale levels:

Small scale: 50 mm to 2,000 mm, with resolution of maximally 1 mm – 5 mm.
Medium scale: 2 m to 50 m, with a resolution in 0.5 cm – 10 cm.
Large scale: larger than 50 m, with a resolution from 0.2 m up to 1 m.

2.1 Multisensory 3D-Modeler

The DLR Multisensory 3D Modeler is a small-scale modeling system with different sensor components [14]. The system's strength lies in multisensory data acquisition. Currently, the system integrates: a laser-range scanner, a texture sensor, a laser-stripe sensor, and a stereo sensor. The laser-range scanner is based on the triangulation principle. Its main features are a low weight, robustness, and its large angle of view. The measurement distance is 50 mm to 300 mm. The texture sensor consists of a single calibrated miniature head camera. The laser-stripe sensor uses a line-laser module in combination with a calibrated miniature head camera, implementing a sensing range from 150 mm to 450 mm. Geometric information at larger distances is acquired with the stereo sensor.

These sensors are integrated into a specially designed, low weight and ergonomic housing (Figure 1). Pose measurement is either done by a passive manipulator (i.e., a robotic arm) or an optical tracking system.[1]

Merging multiple sensors with multiple interfaces is a major problem in sensor synchronization. We chose a two level strategy: First, hardware synchronization allows synchronous measurements. This is implemented by supplying all sensors with a common video synchronization pulse. Secondly, data sets are merged by using the CAN bus as the master software synchronization bus for exchanging timestamps and poses. The acquisition of 3D-data of all sensors is done in the same global coordinate system. The system is very much suitable for digitizing small objects.

2.2 Z+F Imager and DLR Panoramic Camera

This section deals with a description of medium scale sensors (designed at DLR and Z+F) which are already commercially available.

Z+F Imager 5003. The visual laser scanner Imager 5003 of Z+F (see Figure 2) is an optical measuring system based on the transmission of laser light [1]. The laser scanner consists of a one-dimensional (1D) measuring system in combination with a mechanical beam-detection system. Due to the large field of view of the scanner, 360° horizontally (azimuth) and 310° vertically (elevation), a scene to be modeled has to be surveyed from a few points of view only. Besides the 2D intensity information, the Imager 5003 provides additionally 3D range information. Both - intensity and 3D range information - correspond to each pixel. By extracting features in an accurate way, the combination of image processing methods and 3D geometric information is possible. The system itself has different scanning modes, which differ in spatial point distance [from Super High Resolution (20,000 pixel per 360° horizontally and vertically) to Preview (1,275 pixel per 360° horizontally and vertically) mode]. Regarding acquisition time, we report a mode which is popular in industrial environments: 10,000 points horizontally and 5,000 vertically takes 3.22 minutes for a full 360° scan.

DLR Panoramic Camera. A panoramic camera (see left of Figure 3) was developed at DLR Berlin between 1999 and 2001, which allows the acquisition of high-resolution texture maps. A single image is several 100*s* Megapixel, up to multiples of Gigapixel. The camera is basically a rotating CCD line sensor. Three CCD lines (i.e., for the Red, Green, or Blue channel) form a linear CCD array,

[1] See Advanced Realtime Tracking at http://www.ar-tracking.de.

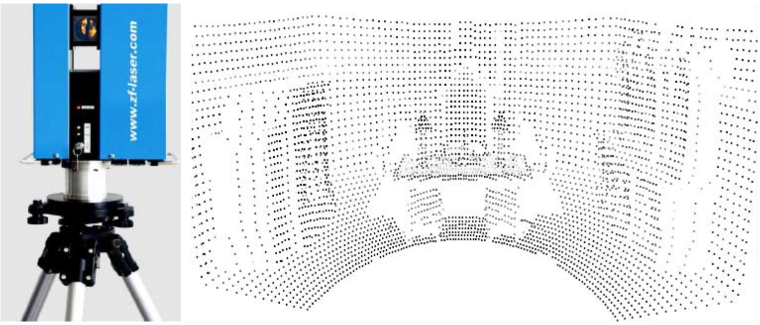


Fig. 2. Z+F Imager 5003 and a point cloud of a scanned room (preview mode)

which is mounted vertically on a focal plane and rotates clockwise, describing a cylindric surface during a full 360° rotation. Scanned data are stored in cylindric coordinates, line by line, and according to the sensor geometry [12]. By using three line CCD chips with 10,200 elements each, very high image resolution can be archived. The resulting image consists of a maximum of 10,200 by 500,000 pixels each containing three 14 Bit RGB values. A typical scan (10,200 × 30,000) using a special optical lens system of 35mm focal length takes about 3 min at daylight, and up to 60 min at dark indoor illumination. The software also includes a package for the geometric and radiometric calibration, which enables the recalculation of the raw data into calibrated images.

Off-Axis Rotation and Principle Angle. If R is set (see Figure 3) to a non-zero value, then the cameras principle point is at *off-axis position*, which is one possibility to acquire stereo images (using different values of ω). The cameras principle point is moving on a circle with radius R. As defined in [6], this circle specifies the *base cylinder*, parameters R and ω are characterized in particular how to optimize these for stereo viewing of a scene characterized by closest and furthest distance between objects of interest and the camera. R and ω are two important parameters of this camera, and their parameter intervals are crucial for specifying the accuracy or flexibility of the camera. For example, the aim might be to have $R = 0$, but it is important to calibrate the actual deviation from this ideal case. Figure 3 shows the DLR panoramic camera and illustrates the parameters R and ω (as studied in PhD projects at CITR [7]).

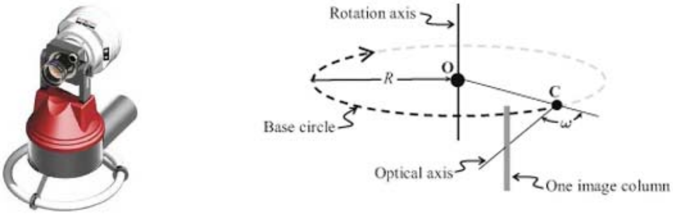


Fig. 3. Panoramic camera and illustration of the camera parameter R and ω

2.3 HRSC - High Resolution Stereo Camera

The High Resolution Stereo Camera (HRSC) has been developed by the DLR Institute of Planetary Research for the exploration of the Martian surface from orbit [15]. The airborne version HRSC-AX is currently used for capturing landscape on Earth, as well as cities from flight altitudes between 1,500 m to 5,000 m. The camera contains nine sensor arrays, which are arranged (in different viewing angles) orthogonally to the flight direction. All arrays have a resolution of 12,000 pixels and record 12 bit per pixel. Five arrays are panchromatic. The other four capture red, green, blue and infrared light. Figure 4 shows the geometry.

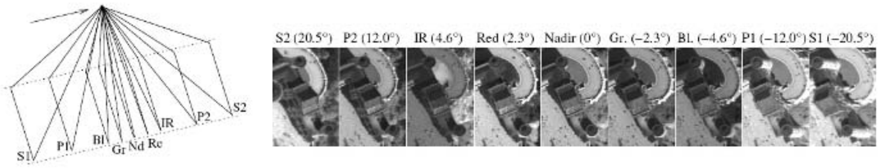


Fig. 4. Basic geometry of HRSC-AX and parts of corrected 2D pushbroom images, with a resolution of 15 cm per pixel

The position and orientation of the camera is continuously measured by a sophisticated GPS/IMU system. Current post-processing includes radiometric corrections as well as refinements of all camera positions, orientations and time offsets by means of photogrammetric methods based on HRSCs multi-stereo image information [13]. A geometric correction step projects the pixels of each array at all camera positions onto an artificial plane, resulting in nine 2D images, in which effects caused by high or low frequency orientation variations are eliminated, while disparities caused by terrain and buildings still remain. Epipolar lines are “almost” straight parallel lines in these images. The reason for “almost” is the (in general) non-linear flight path of the camera. The resulting 2D images and the inherent disparity ranges are typically huge (e.g. several 100 MPixel with 1,000 pixel disparity range).

3 Principles and Methods for 3D Modeling

This section elaborates on principles and methods for generating 3D models. Preprocessing of laser data is neglected in this brief note. A method for 3D surface reconstruction, suitable for arbitrary sized raw data sets, is outlined. Then, the task of texture mapping is addressed (needed because color information is required in the context of cultural heritage preservation or visualization). Medium scale objects require that the sensor system is relocated for full coverage. Therefore, these data sets from different view points need to be matched [10]. Finally, stereo matching for the creation of large scale models is discussed.

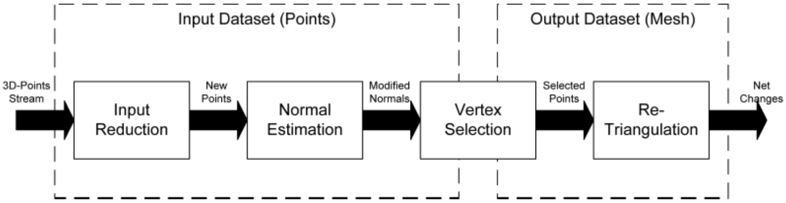


Fig. 5. Reconstruction pipeline

3.1 Surface Model Generation

In order to generate models from different modeling systems, an algorithm is needed that is not specific to the sensor's data format. Furthermore, it has to cope with large input data sets. Therefore, an online triangulation tool is used, that is able to generate and improve a triangle mesh directly from unorganized 3D sets of points. The algorithm processes sensor data by incremental insertion of 3D points, which suits generic 3D sensors. The reconstruction pipeline is divided into four steps, input reduction, normal estimation, vertex selection, and re-triangulation, as illustrated in Figure 5. The input points and the vertices of the mesh are stored into separate point sets, each implemented in a hierarchical data structure that allows fast insertion and search of local point neighborhoods. The implemented structure only requires a small memory overhead, so it is perfectly suited for very large data sets [2].

3.2 Mesh Optimization

The quality of the mesh, generated by the described method, is improved as follows:

(i) *Filters:* Basically we consider two types of filters for improving 3D data. Firstly, the system specific noise (e.g., measuring noise of the LRF) is reduced by various filters (e.g., median filter, histogram filters, spike filters, Gaussian pyramids, etc.), before we turn to the meshing algorithm. Errors not caused by the system noise (e.g., errors caused by an unfavorable incident angle) and known object geometry are expected. They are fixed subsequent to generating the first initial dense mesh in a second filtering process. This is, in fact, the step of repairing the mesh.

(ii) *Filling Holes:* Reasons for visible holes in triangle meshes are either missing or false triangulation. If there are vertices with false normals or triangles with false triangulation orientation, then they will be displayed as holes. Therefore, the task is divided into filtering out "illegal" triangles, and filling by a recursive algorithm considering the 3D relation between vertices and edges.

(iii) *Mesh Reduction:* The normal vectors of the surrounded triangles of each vertex are analyzed. If their difference is below a certain threshold, the vertex is deleted and the resulting hole is newly triangulated. (Experiments showed that about 90% of triangles and vertices are reduced when reconstructing Neuschwanstein Castle.)

(iv) *Smoothing:* Because the topology is changed after mesh reduction, we use a scale depended fairing algorithm to smooth the mesh.

3.3 Texture Mapping

Texture mapping adds a (photo-)realistic impression to a given 3D model by linking each of its surface patches with an image, called texture. Often, predefined synthetic textures are used for 3D models. Here, the real texture of the model is gathered by moving a camera around the object in question. The sequence of texture images is then subsequently mapped to the real 3D model acquired by the scanning device. The texture mapping process requires known intrinsic (including distortion) and extrinsic camera parameters relative to the 3D model. This "image-to-model" registration is accomplished by measuring, i.e. tracking the camera position in a world coordinate system, or by photogrammetric determination of those. By knowing the exact position and direction of the camera, the images can be projected onto the 3D-model. The determination of texture coordinates for off-axis rotating cameras corresponding to a 3D model is described in detail in [9]. The texture mapping for integrated cameras, as used for the 3D-modeler is shown in [5], and for the Imager 5005 in [1].

3.4 Stereo Matching by Semi-Global Matching (SGM)

Corrected 2D pushbroom images can be used for stereo reconstruction. The required stereo matching method has to be efficient for operating on huge images and disparity ranges. Furthermore, stereo matching must be accurate for maintaining sharp object boundaries that are common in the anticipated scenes of urban areas. Hierarchical, correlation-based stereo methods are often used in these scenarios [15], due to their efficiency. However, these approaches are well known for blurring sharp object boundaries [3]. However, global methods are typically slow and memory intensive, which makes them unsuitable for the anticipated application. The problem of accurate and efficient stereo matching is solved by the Semi-Global Matching (SGM) method [4]. SGM aims to determine the disparity image D, such that the cost $E(D)$ is a minimum.

$$E(D) = \sum_{\underline{p}} C(\underline{p}, D_{\underline{p}}) + \sum_{\underline{q} \in N_{\underline{p}}} P_1 T[|D_{\underline{p}} - D_{\underline{q}}| = 1] + \sum_{\underline{q} \in N_{\underline{p}}} P_2 T[D_{\underline{p}} - D_{\underline{q}} > 1]$$

This cost function evaluates pixelwise matching costs $C(\underline{p}, D_{\underline{p}})$ at the pixel $\underline{p}$ with the disparity $D_{\underline{p}}$. Piecewise smoothness of the disparity image is supported by adding a small cost P_1 for all small disparity changes and a higher cost P_2 for all higher disparity changes. Adding a constant cost for all higher disparity changes preserves discontinuities. Finding the minimum of this energy is an NP-complete problem. The SGM algorithm approximates the global minimization by pathwise minimizations from all directions.

The complexity is only $O(ND)$ like correlation based approaches, but the memory consumption is also proportional to ND (= number of pixels times the disparity range). The pixelwise matching cost $C(\underline{p}, D_{\underline{p}})$ is based on hierarchically computing Mutual Information (MI) instead of intensity differences. This makes it robust against recording differences and illumination changes, which is possible since pushbroom cameras capture corresponding points at different times (during the flight).

The SGM method has been adapted for matching huge HRSC images. Firstly, the generally "curved epipolar lines" of aerial pushbroom images are explicitly calculated in contrast to other methods [15]. This reduces the disparity range from a 2D area to a 1D segment of the epipolar line and saves run time as well as memory. Secondly, the huge images are split into manageable tiles for matching. Tiles are defined slightly overlapping, and pixels near image borders are rejected, because they receive support only from one side by the global cost function. Thirdly, multi-baseline matching [4] is used for matching the five panchromatic images of the HRSC, weighted by their recording angle. Optionally, the red and green images are also matched against the panchromatic nadir image, which is possible with MI matching. Finally, the disparity range is determined automatically, by first processing downscaled images (e.g., by factor 16) with a very large disparity range that is known to cover all situations. A reduced range is determined from the result and used for higher resolutions. The disparity range determination is done during the hierarchical computation of MI. The matching result is used for calculating Digital Elevation Models (DEM), and subsequently for the generation of true ortho-images based on the DEM.

4 Applications

This section illustrates applications for small (busts, using 3D Modeler), medium (castle Neuschwanstein, using Z+F Imager and panoramic camera), and large scale (the city of Berlin, using HRSC).

4.1 Small Scale Models

An example of a 3D Modeler result was shown in Figure 1 (right). Here, the focus is on hand-guided acquisition. The operator sweeps the system manually over the surface of the object, and the 3D-model is reconstructed simultaneously by the surface generation algorithm in the above section. Immediate visual feedback helps the user during the process of a complete digitization of the object; see [2] for details. Images from the texture sensor are integrated into a visual feedback.

4.2 Medium Scale Models

The digitization of historic buildings is an example in this category. On the historic site of castle Neuschwanstein near Füssen, Germany, maps and views of approximately 450 rooms of the castle (see Figure 6) have been generated. Multiple high resolution scans at several positions in the rooms are performed. Number of scanner positions depends on the size and the complexity of each room (i.e., occlusions by objects like stone columns).

Preprocessing steps are applied before the data of a room can be merged to a single 3D model. First, the raw scanner data is transformed into equidistant grids (i.e., to an image with equidistant pixels) using the intrinsic parameters. Then the camera transformation using the camera's intrinsic parameters (distortion, scale, color shift, and white balance) is applied to every picture, resulting in calibrated cylindrical images; see Figure 6 (right).

Afterwards, the extrinsic parameters (i.e. camera and scanner positions and orientations, as well as range scale, if necessary) are estimated. Now, all scans of a room are transformed to a 3D point cloud in the world coordinate systems. The 3D points are triangulated using the surface generation tool from Section 3.1. Afterwards, a mesh optimization process (hole filling algorithm and mesh reduction from Section 3.2 are applied to further improve the mesh; see results in Figure 7. Finally, the 3D model is textured with either the intensity image of the scanner or the high resolution color image of the panoramic camera.

Starting in 2002, approximately $9 \cdot 10^{10}$ 3D-points in 1,800 scans of the Z+F Imager have been acquired in multiple campaigns. The accuracy of the scanner data after adjustment is about 2mm. A 3D-model of the King's office in castle Neuschwanstein is shown in Figure 7 whereas the figure (bottom-right) illustrates the same model textured using panoramic camera images and intensity images from the LRF itself.

4.3 Large Scale Models

Cities and landscapes have been reconstructed using the SGM stereo method (Section 3.5), applied to pre-processed (Section 2.3) HRSC images. 3D visualizations are created directly from DEMs using ortho-images as top-view texture;

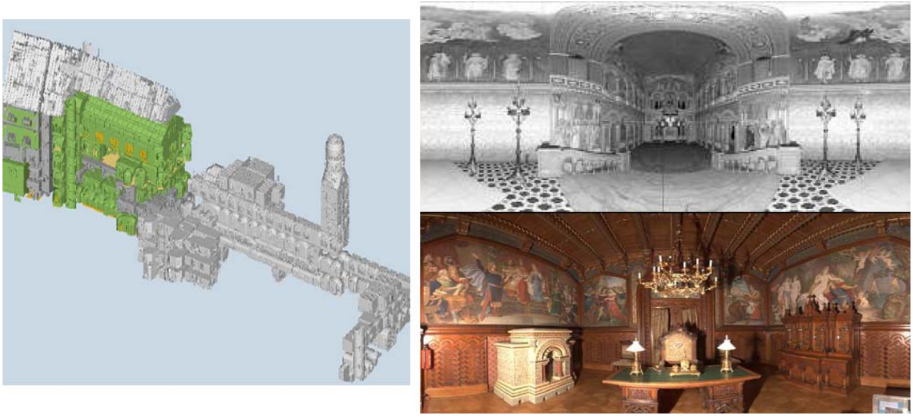


Fig. 6. Left: 450 rooms of Neuschwanstein Castle have been scanned with approximately $9 \cdot 10^{10}$ 3D-points in 1,800 scans of the Z+F Imager. Right: a laser scan of the calibrated Z+F Imager 5003 (local polar coordinate system 360° × 180°), and a cylindrical calibrated panoramic image.

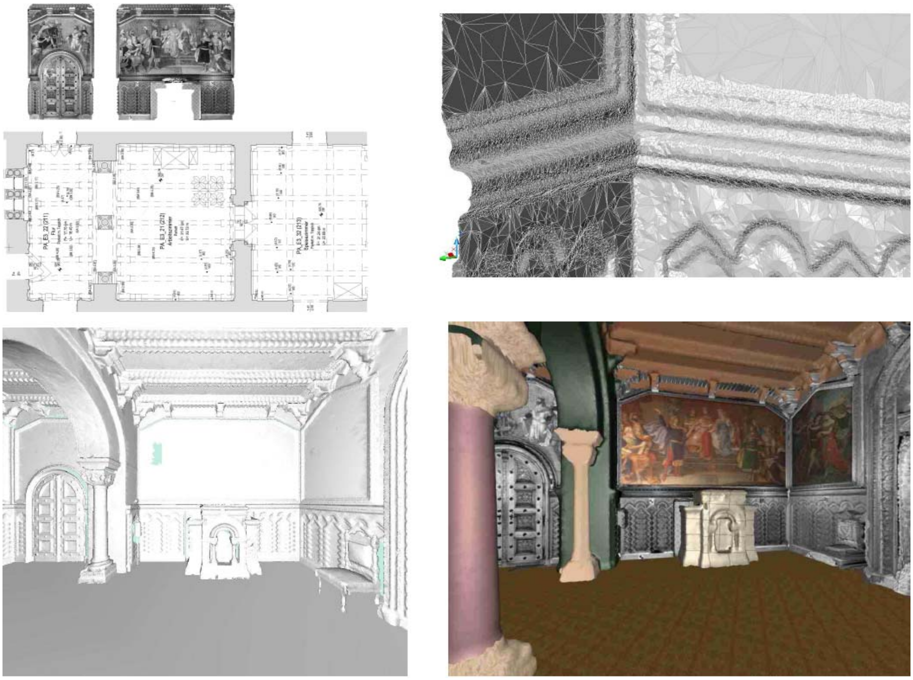


Fig. 7. Floor plan of Ludwig II's office, with orthophotos (top-left), a detail of a room corner (top-right), the 3D-model (bottom-left), and the textured model (bottom-right)

see Figure 8. Additionally, side-view textures of buildings and other objects are taken from the forward and backward looking stereo images; see Figure 4. For a better quality of side-view textures, tests of data fusion with the panoramic camera and the HRSC have started; see Figure 9.

The whole process of stereo matching, DEM, ortho-image, side-view texture creation and visualization is fully automatically. A 110 km^2 area of the city of Berlin (see Figure 8) has been recorded in 6 parallel, partly overlapping flights at an altitude of 4,100 m over ground. The images were scaled for a ground resolution of 20 cm per pixel, although the true resolution is less, due to the high

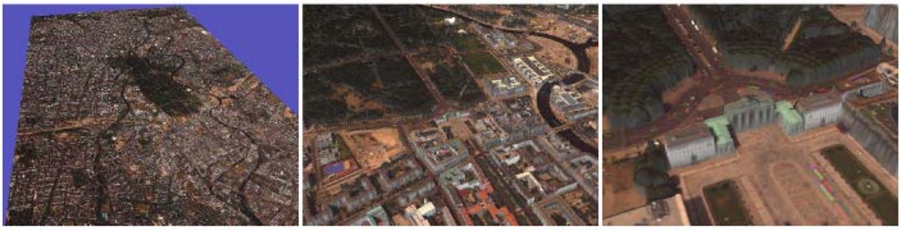


Fig. 8. Automatically generated 3D model of Berlin (different zoomings)

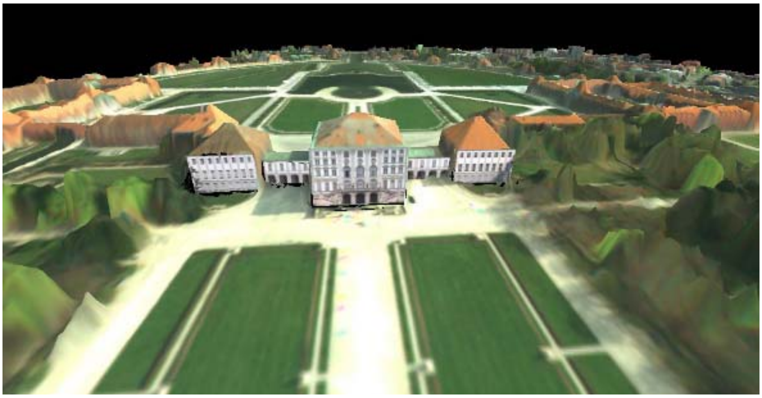


Fig. 9. Extracted 3D model of the DEM data (HRSC) with (partially) mapped textures captured by a terrestrial panoramic camera

recording altitude. Each flight contributed approximately 1 billion height values and pixels to the DEM and ortho-image, which have a total size of around 2.7 billion values. The total processing time was 18 days on a 2.8 GHz Xeon computer.

5 Conclusions and Future Work

This paper briefly presented all aspects for recording, modeling and visualization of cultural heritage. We stressed the methodical similarity between 3D modeling in robotics and or cultural heritage. Systems, algorithms and results were presented for all scales of objects as defined in Section 2. For the acquisition of small scale objects, a 3D modeling framework was developed, integrating texture mapping, mesh optimization, hole filling, and a generic sensor interface, as well as providing visual feedback on surface reconstruction and color (texture) acquisition. Future research is directed towards the improvement of algorithms and sensor systems, in respect to accuracy and efficiency in generating photo-realistic 3D-models. Currently, modeling of medium scale objects still involves manual interference. The mesh optimization process needs to be simplified. Future research in stereo processing deals with a comparison of DEM's against ground truth and DEM's from other sources. Large scale model generation requires high-performance computing, motivating the implementation of methods on a processing cluster. A cluster of 12 double processor computers will finally reduce the computation of huge areas (e.g. 400 km^2) to reasonable times such as a few days only.

Acknowledgements. The authors thank Ulrich Hagn, Franz Hacker, Klaus Strobl and Eric Wahl for supporting the 3D-Modeler activities, as well as Johann Heindl for contributions to the Neuschwanstein project. Thanks to Frank Scholten for pre-processing HRSC images, and to Tom Abmayr and the staff at Z+F for excellent cooperation, and to Tim Bodenmüller for their work on meshing algorithms. We are indebted to Reinhard Klette for his support and PhD supervision in research on panoramic topics.

References

1. T. Abmayr, F. Härtl, M. Breitner, M. Ehm, and C. Fröhlich. Multimodale Sensorfusion auf Basis des Imager 5003. *Photogrammetrie, Laserscanning, optische 3D-Messtechnik, Beiträge der Oldenburger 3D Tage*, 2006.
2. T. Bodenmüller and G. Hirzinger. Online surface reconstruction from unorganized 3D-points for the DLR hand-guided scanner system. In *Symp. 3D Data Processing, Visualization, Transmission*, Thessaloniki, Greece, 2004.
3. H. Hirschmüller, P.R. Innocent, and J.M. Garibaldi. Real-time correlation-based stereo vision with reduced boarder errors. *IJCV*, **47**:229–246, 2002.
4. H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Conf. Computer Vision Pattern Recognition*, pages II:807–814, San Diego, USA, 2005.
5. G. Hirzinger et al. Photo-realistic 3D-modelling - From robotics perception towards cultural heritage. *Int. Workshop Recording Modelling Visualization Cultural Heritage*, Ascona, Switzerland, 2005.
6. F. Huang, S.-K. Wei, R. Klette, G. Gimel'farb, R. Reulke, M. Scheele, and K. Scheibe. Cylindrical panoramic cameras - from basic design to applications. In *Image Vision Computing New Zealand*, 101–106, 2002.
7. F. Huang, S.-K Wei, and R. Klette. Geometrical fundamentals of polycentric panoramas. ICCV 01, pages I:560-565, Vancouver, Canada, July, 2001.
8. R. Klette and R. Reulke. Modeling 3D scenes: Paradigm shifts in photogrammetry, remote sensing and computer vision. CITR-TR-155, The University of Auckland, 2005.
9. R. Klette and K. Scheibe. Combinations of range data and panoramic images - new opportunities in 3D scene modeling. In *Computer Graphics Imaging Vision: New Trends*, pages 3–10, 2005.
10. R. Liu and G. Hirzinger. Marker-free automatic matching of range data. *2nd Panoramic Photogrammetry Workshop*, 2005.
11. W. Niemeier. Einsatz von Laserscannern für die Erfassung von Gebäudegeometrien. *Geäudeinformationssystem*, **19**:155–168, 1995.
12. K. Scheibe, H. Korsitzky, R. Reulke, M. Scheele, and M. Solbrig. Eyescan - a high resolution digital panoramic camera. In *Robot Vision*, pages 77–83, New Zealand, LNCS, Springer, 2001.
13. F. Scholten and K. Gwinner and F. Wewel. Angewandte digitale Photogrammetrie mit der HRSC-AX. *Photogrammetrie Fernerkundung Geoinformation*,**5**:317–332, 2002.
14. M. Suppa and G. Hirzinger. A novel system approach to multisensory data acquisition. In *Intelligent Autonomous Systems IAS-8*, Amsterdam, The Netherlands, 2004.
15. F. Wewel, F. Scholten, and K. Gwinner. High resolution stereo camera (HRSC) - multispectral 3D-data acquisition and photogrammetric data processing. *Canadian J. Remote Sensing*, **26**:466–474, 2000.

Boundary Based Orientation of Polygonal Shapes

Joviša Žunić*

Computer Science Department, Exeter University
Harrison Building, Exeter EX4 4QF, U.K.
J.Zunic@ex.ac.uk

Abstract. The computation of a shape's orientation is a common task in many areas of computer vision and image processing applications. It is usually an initial step or a part of data preprocessing. There are several approaches to the problem – most of them could be understood as the 'area based' ones. In spite of many unavoidable problems where working with shape boundaries in discrete space, the demand for a pure 'boundary based' method, seems to be very reasonable. Such a method for shapes having polygonal boundaries is presented in this paper. We define the shape orientation by the line that maximises the total sum of squared lengths of projections of all the shape boundary edges onto this line. Advantages and disadvantages of the method are discussed.

Keywords: Shape, orientation, image processing, early vision.

1 Introduction

Many image processing and shape analysis tasks start with a normalisation procedure [4,5,6]. For a successful application (in robotics, medical imaging, industry inspection tasks, etc) it is important that the reference frame is properly determined. Shape position and orientation define the frame of reference. Usually, the shape position is defined by its gravity center and that is a very common approach. On the other side, the computation of orientation is not a straightforward task and there are many approaches in defining the shape orientation.

Due to the variety of shapes as well as the diversity of applications there is not a single method for computing the shape orientation that could be successfully applicable to all shapes. For that reason, several methods have been developed ([1,3,7,8,12,13,14]). Different techniques have been used, including those based on geometric moments, complex moments, and principal component analysis, for example. Suitability of those methods strongly depends on the particular situation in which they are applied, as they each have their relative strengths and weaknesses.

The majority of existing methods for the computing orientation are 'area based' – i.e. the computation takes into account all the points that belong to

* The author is also with the Mathematical institute of Serbian Academy of Sciences, Belgrade.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 108–117, 2006.

the shape, not only the boundary points. Among area based methods, the most standard one says that shape orientation is determined by its axis of the least second moment of inertia ([4,5,6]). The axis of the least second moment of inertia of a shape is defined as a line that minimises the integral of the squared distances of the shape points to the line. When working in discrete space where shapes are represented by finite sets of points (set of pixels, for example) then the 'integral' should be replaced with the 'sum'. Obviously, the method is motivated very naturally and is simple to compute in both ('real' and 'discrete') versions.

Because the standard method is area based it is very robust with respect to noise and boundary defects. The problem is that there are many situations where the method does not give any answer what the shape orientation should be. There are many regular and irregular shapes where this standard method does not work ([14,15]). Also, in many situations the robustness of a method is a desirable property, but sometimes it could be a disadvantage (in high precision inspection tasks, for example). Further, it could happened that some shapes are "nonorientable" (see [16]) by the standard method, but they could be easily oriented if narrow intrusions or scribble details on them exist. Those details correspond to a relatively small percentage of pixels (when working with digital images) and are not detectable by robust methods. In this paper we present a method where the orientation is computed based on the shape boundary. Consequently the method could overcome some of the mentioned problems. In a typical situation, the new method takes into account the complete boundary – not only parts belonging to the convex hull of the considered shape, as in [2,9], for example. But the method can be applied to shapes whose boundaries are partially detected and to shapes where scrabble details are considered as the boundary parts.

The paper is organised as follows. Section 2 introduces the new method and analyses its basic properties. Examples and related comments are in Section 3. Concluding remarks are in Section 4.

2 Boundary Based Shape Orientation

In this section we define a new method for computing the orientation of shapes with polygonal boundaries. The method is boundary based and takes into account all the boundary points. Let us mention that there are naive methods that are also boundary based. For example, the orientation of a polygonal shape having vertices P_1, $P_2, \ldots,$ and P_n could be defined as the average value of the angles between the edges P_iP_{i+1} ($1 \leq i \leq n$ and $P_{n+1} \equiv P_1$) and the x-axis. The method is extremely simple but it has many disadvantages. From such a definition, each two shapes whose edges make identical angles with the x axis must have the same computed orientation. But this is not a desirable property. Edges of polygonal shapes presented on Fig.1 make identical angles with the x-axis but their assigned orientations should be different.

It is not a surprise that such a trivial method would not give good results. It is pretty presumable that the edge lengths have to play a significant role in the

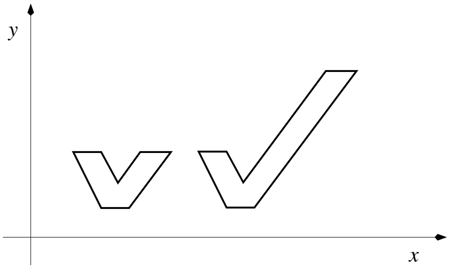


Fig. 1. The edges of presented polygons make identical angles with the x-axis but their assigned orientations should be different

orientation definition. Here we define the orientation of a polygonal shape by the direction that maximises the total sum of the squared lengths of the projections of all boundary edges onto a line defined by this direction – see Fig.3 for an illustration. We give the following formal definition.

Definition 1. *Let a shape with a polygonal boundary P. The orientation of the shape is defined by the angle $\alpha = \alpha_0$ for which the total sum*

$$F(\alpha, P) = \sum_{e \text{ is an edge of } P} |\mathbf{pr}_\alpha(e)|^2 \tag{1}$$

of squared lengths of projections of edges of P onto a line having the slope α reaches its maximum.

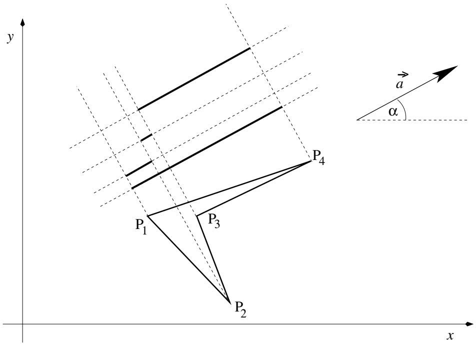


Fig. 2. Projections of the edges of the polygonal shape (having vertices $P_1,\ P_2,\ P_3, P_4$) onto lines having the slope α are presented

It is an adventage that the new definition is motivated naturally. Also, in canonical cases, where the orientation of polygonal shapes seems to be very distinct, the method gives expected results. For instance, the orientation of a rectangle is expected to be coincident with its longer edge, while the orientation of a very elongated triangle having exactly one axis of symmetry should be coincident with such a symmetry axis, etc. That is exactly what happens if the new method is applied. Without loss of generality we can assume that an edge of the considered rectangle and one of the edges of the considered triangle are parallel to the x-axis – for notations we reffer to Fig.3.

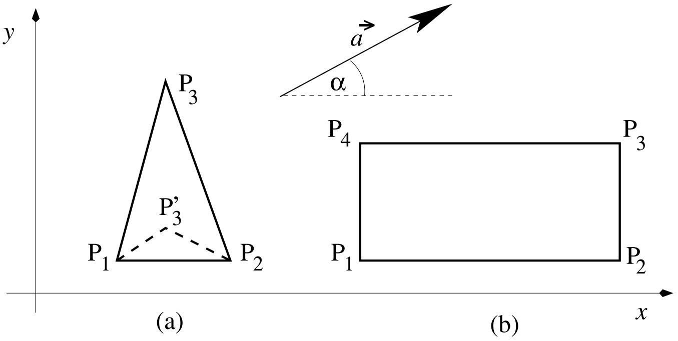


Fig. 3. The computed orientation of $\triangle P_1P_2P_3$ is 90^o. The computed orientation of $\triangle P_1P_2P_3'$ is 0^o. The computed orientation of the rectangle $P_1P_2P_3P_4$ is 0^o.

- For the rectangle P having the vertices P_1, P_2, P_3, and P_4, let $|P_1P_2| = |P_3P_4| = p$ and $|P_2P_3| = |P_4P_1| = q$. The sum $F(\alpha, P)$ of the squared lengths of projections of the edges onto a line having the slope α is

$$F(\alpha, P) = 2 \cdot p^2 \cos^2 \alpha + 2 \cdot q^2 \sin^2 \alpha = 2 \cdot (p^2 - q^2) \cdot \cos^2 \alpha + 2 \cdot q^2.$$

Consequently:

  - If $p > q$ then the maximum of $F(\alpha, P)$ is $2 \cdot p^2$ and it is reached for $\alpha = 0$, i.e., the rectangle is oriented in accordance with the longer edges;
  - If $p < q$ then the maximum of $F(\alpha, P)$ is $2 \cdot q^2$ and it is reached for $\alpha = \pi/2$. Again, the rectangle is oriented in accordance with the longer edges;
  - If $p = q$ then the rectangle degenerates into a square and the method does not suggest what the orientation should be. The sum of the squared lengths of projections of all the edges is the same for all α, i.e. $F(\alpha, P) = constant$.

- For the triangle $T = \triangle P_1P_2P_3$ let $\beta = \angle(P_2P_1P_3) = \angle(P_1P_2P_3))$. The sum $F(\alpha, T)$ of the squared lengths of projections of all the edges of T onto a line having the slope α is

$$F(\alpha, P) = q^2 \cdot \cos^2 \alpha + p^2 \cdot \cos^2(\beta + \alpha) + p^2 \cdot \cos^2(\beta - \alpha)$$

where p denotes the length of the edges P_3P_1 and P_2P_3 while q denotes the length of P_1P_2. Taking into account $\cos \beta = \dfrac{q}{2p}$ and by using elementary transformations $F(\alpha, P)$ can be expressed as

$$F(\alpha, P) = (q^2 - p^2) \cdot \cos^2 \alpha + p^2 \cdot \left(1 - \frac{q^2}{4 \cdot p^2}\right).$$

So,

- If $q < p$ then the maximum of $F(\alpha, P)$ is reached for $\alpha = \pi/2$ – i.e. the computed orientation coincides with the axis of symmetry;
- If $q > p$ then the maximum of $F(\alpha, P)$ is reached for $\alpha = 0$ and the computed orientation is orthogonal to the axis of symmetry; The obtained orientation is debatable if p is close to q, but it is very acceptable if q is much bigger than p. Particularly, in the limit case when $p \to q/2$ the triangle degenerates into a horizontal line segment whose measured orientation should be 0 degrees (as computed by the method);
- If $q = p$ then $F(\alpha, P)$ is a constant function and does not depend on α. Thus, the method does not tell what the orientation should be.

It is worth to mention that the exactly same orientations are obtained if $\triangle P_1P_2P_3$ is oriented by the standard method.

In the previous two simple cases the orientation was easy to compute. The question is: *Is the orientation easy to compute in the case of an arbitrary polygonal area?* We will show that the method can be applied easily to all polygonal shapes. Even more, it could be applied to not necessarily closed polygonal lines, what can be of an importance if working with incomplete data, i.e. if some boundary parts are missed or not extracted properly. We proceed with the following theorem.

Theorem 1. *Let an n-gon P with edges e_i, $i = 1, \dots, n$. Also, let α_i denote the angle between e_i and the x-axis. If the total sum*

$$\sum_{i=1}^{n} |\mathbf{pr}_\alpha(e_i)|^2$$

of the squared lengths of projections of the edges e_i onto a line having the slope α reaches its maximum for $\alpha = \alpha_0$ then

$$\tan(2 \cdot \alpha_0) = \frac{\sum\limits_{i=1}^{n} |e_i|^2 \sin(2\alpha_i)}{\sum\limits_{i=1}^{n} |e_i|^2 \cos(2\alpha_i)}. \tag{2}$$

Proof. Let e_i and α_i $(1 \le i \le n)$ are as in the statement of the theorem. The length $|\mathbf{pr}_\alpha(e_i)|$ of the projection of the edge e_i onto a line having the slope α is

$$|\mathbf{pr}_\alpha(e_i)| = |e_i| \cdot |\cos\alpha_i \cos\alpha + \sin\alpha_i \sin\alpha| = |e_i| \cdot |\cos(\alpha_i - \alpha)|$$

and the function that should be maximised is

$$F(\alpha, P) = \sum_{i=1}^{n} |\mathbf{pr}_\alpha(e_i)|^2 = \sum_{i=1}^{n} |e_i|^2 \cos^2(\alpha_i - a). \tag{3}$$

The maximum of $F(\alpha, P)$ can be computed on a standard manner. The first derivative $dF(\alpha, P)/d\alpha$ can be expressed as

$$\begin{aligned}\frac{dF(\alpha,P)}{d\alpha} &= -\sum_{i=1}^{n} |e_i|^2 \sin(2\alpha_i - 2\alpha) \\ &= \sum_{i=1}^{n} |e_i|^2 (\cos(2\alpha_i)\sin(2\alpha) - \sin(2\alpha_i)\cos(2\alpha)). \qquad (4)\end{aligned}$$

Setting $dF(\alpha, P)/d\alpha = 0$ we obtain that the angle α_0 where $F(\alpha, P)$ reaches its maximum satisfies (2) . This establishes the proof. □

Now we give three remarks that follow directly form the proof of Theorem 1. Those remarks clarify situations where the method can be applied.

Remark 1. Since $F(\alpha, P)$ is a continuous function it reaches its extreme values on the closed interval $[0, 2\pi]$. For each given polygon P those extreme values are easy to compute (in accordance with (2)). Obviously, if the maximum is reached at $\alpha = \alpha_0$ then the minimum is reached at $\alpha = \alpha_0 + \pi/2$.

Remark 2. Due to the simplicity of the method, it is expected that there are situations where the method does not give an answer to what the shape orientation should be. By the way, it was already shown in the case of a regular triangle and in the case of a square. Now we can give a formal characterisation of shapes that cannot be oriented by the new method. Looking at (4) we can see that for each n-gon P with

$$\sum_{i=1}^{n} |e_i|^2 \cos(2\alpha_i) = 0 \qquad \text{and} \qquad \sum_{i=1}^{n} |e_i|^2 \sin(2\alpha_i) = 0 \qquad (5)$$

the first derivative $dF(\alpha, P)/d\alpha$ is identically equal to zero. Further, this implies that $F(\alpha, P)$ is constant and consequently, it does not suggest any particular direction as the orientation of P.

Remark 3. Theorem 1 holds if P is an arbitrary polygonal curve (not necessarily a closed polygon). The proof does not need any modification.

3 Discussion and Some Examples

In this section we illustrate how the method works in practice. For each shape presented on Fig.4 both, orientation computed by the new method and orientation computed by the standard method (the numbers in the brackets) are given. It is obvious that in the case of essential intrusions or in the case of long thin details a big difference between two computed orientations is possible. For instance, the new method gives that the orientation of the sketch of rabbit strongly depends on the position of its ears – such an impact is lower if the standard method is applied (see the first column). Also, the impact of the trunk position is much higher if the sketch of elephant is oriented by the new method than if it is oriented by the standard method (see the second column). The change in the intrusion position of the shape from the third column cannot be detected by the standard method, while such a change has a big impact on the computed orientation if the new method is applied. The last two shapes from the first row, as well as the last shape in the second row, have reasonable computed orientations in the sense of both methods. The sketch of Africa has not distinct orientation and that is a reason for so big difference in computed orientations.

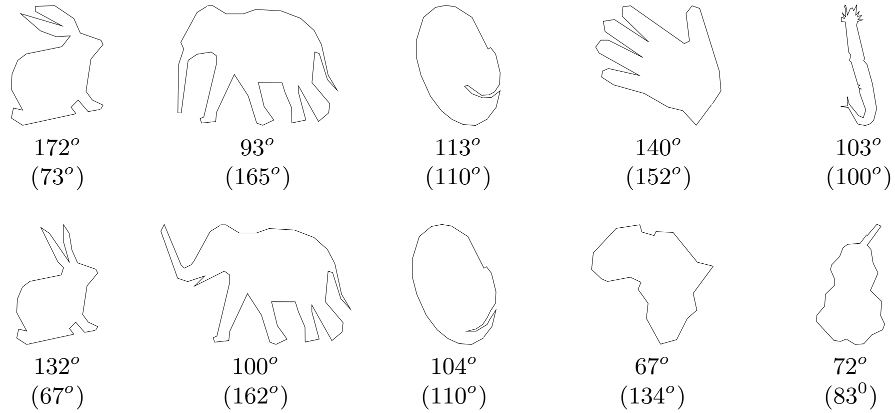


Fig. 4. Computed orientations by the new method. Orientations computed by the standard method are in brackets.

Since the new method is boundary based, it has to be very sensitive to the boundary defects caused by a noise or by boundary defects, for example. On Fig.5 a shape is presented in order to illustrate possible noise effects to the computed shape orientation. Some noise effects can be corrected by a suitable choice of polygonal approximation (shape in the middle), but once again, big boundary defects (third shape) must to lead to an essential change in the computed orientation. The change in the computed orientation if the standard method is applied (the numbers in brackets) are much smaller (as expected) since the method is area based, what implies its robustness.

As stated in Remark 3, objects composed by one or more (not necessarily closed) polygonal lines can be oriented by the new method. All the edges that belong to the appearing poly-lines must be taken into account and the orientation is determined with the direction that maximises the total sum of squared lengths of the projections of those edges onto a line coincident with this direction. Of course, the area based methods (e.g the standard method) are not directly applicable to such objects. A few examples are given on Fig. 6. The first object is a poly-line by its nature. The second and third objects consist of several poly-line parts – also by their nature. The last shape presents a microorganism

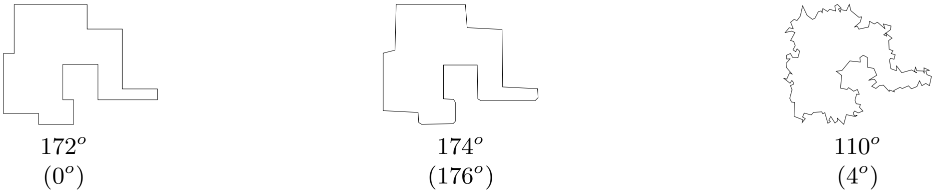


Fig. 5. Noise effects illustration

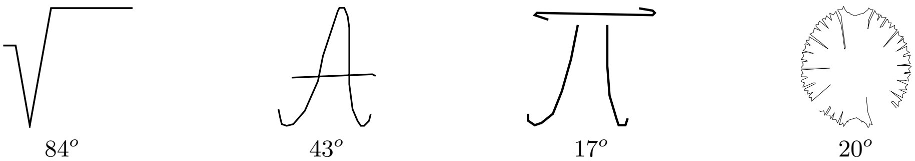


Fig. 6. Polygonal line orientations computed by the new method

whose boundary is not extracted completely due to the fact that there was not an essential contrast between the object pixels and the pixels that correspond to the background .

At the end of this section we prove a very desirable property of the new method which preserves that the computed orientation of reflective symmetric shapes is either consistent or orthogonal to their symmetry axes. The result is in accordance with the earlier discussion related to the triangles from Fig.3.

Lemma 1. *Let a reflective symmetric polygonal shape P whose symmetry axis has the slope β. Then the function $F(\alpha, P)$ reaches its maximum (minimum) either for $\alpha = \beta$ or for $\alpha = \beta + \pi/2$.*

Proof. Without loss of generality we can assume that $\beta = 0$ i.e., P is reflective symmetric with respect to the x-axis. The edges $e_1, e_2, \ldots, e_n$ of P can be divided into two disjoint groups that belong to two half planes determined by the x-axis. (If an edge intersects x-axis it should be split onto two parts each one belonging to the opposite half planes.)

Let $e'_1, e'_2, \ldots, e'_m$ be edges lying above x-axis and let α'_i be the corresponding angles between those edges and the x-axis.

Also, let $e''_1, e''_2, \ldots, e''_m$ be edges lying below the x-axis and let α''_i be the corresponding angles between those edges and the x-axis. Since P is symmetric with respect to the x-axis, we have

$$|e'_i| = |e''_i| \quad \text{and} \quad \alpha''_i = 180 - \alpha'_i \qquad \text{for all} \quad i = 1, 2, \ldots, m.$$

Then,

$$F(\alpha, P) = \sum_{e_i \ is \ an \ edge \ of \ P} |e_i|^2 \cos^2(\alpha_i - \alpha)$$

$$\begin{aligned}
&= \sum_{i=1}^{m} |e_i'|^2 \cos^2(\alpha_i' - \alpha) \quad + \quad \sum_{i=1}^{m} |e_i''|^2 \cos^2(\alpha_i'' - \alpha) \\
&= \sum_{i=1}^{m} |e_i'|^2 \left(\cos^2(\alpha_i' - \alpha) + \cos^2(180 - \alpha_i' - \alpha)\right) \\
&= 2 \cdot \sum_{i=1}^{m} |e_i'|^2 \left(\cos^2 \alpha_i' \cdot \cos^2 \alpha + \sin^2 \alpha_i' \cdot \sin^2 \alpha\right) \\
&= 2 \cdot \sum_{i=1}^{m} |e_i'|^2 \sin^2 \alpha_i' + 2 \cdot \sum_{i=1}^{m} |e_i'|^2 \left(\cos^2 \alpha_i' - \sin^2 \alpha_i')\right) \cdot \cos^2 \alpha \\
&= \sum_{e_i \ is \ an \ edge \ of \ P} |e_i|^2 \sin^2 \alpha_i \quad + \\
&\quad \sum_{e_i \ is \ an \ edge \ of \ P} |e_i|^2 \left(\cos^2 \alpha_i - \sin^2 \alpha_i\right) \cdot \cos^2 \alpha.
\end{aligned}$$

So, we distinguish three situations:

(1) $\sum_{e_i \ is \ an \ edge \ of \ P} |e_i|^2 \left(\cos^2 \alpha_i - \sin^2 \alpha_i\right) < 0$

then $F(\alpha, P)$ reaches its maximum for $\alpha = 0$. The minimum is reached for $\alpha = \pi/2$. (Note: The symmetry axis corresponds to the computed shape orientation.)

(2) $\sum_{e_i \ is \ an \ edge \ of \ P} |e_i|^2 \left(\cos^2 \alpha_i - \sin^2 \alpha_i\right) > 0$

then $F(\alpha, P)$ reaches its minimum for $\alpha = 0$. The maximum is reached for $\alpha = \pi/2$. (Note: The symmetry axis is orthogonal to the computed shape orientation)

(3) $\sum_{e_i \ is \ an \ edge \ of \ P} |e_i|^2 \left(\cos^2 \alpha_i - \sin^2 \alpha_i\right) = 0$

then $F(\alpha, P)$ is a constant function. The minimum and maximum are the same and reached at each point – what is also (formally speaking) in accordance with the statement of the lemma. □

4 Concluding Remarks

In this paper we focused on the orientation of shapes presented by polygonal boundaries. That is not a strong restriction. Indeed, In computer vision applications we work with discrete data and consequently there is always an inherent loss of information. Many difficulties would appear if trying to recover information about boundaries of original shapes. The curvature computation as well as the curve length estimation (from the corresponding discrete data) are already know as very difficult problems. On the other hand there are many efficient algorithms for polygonal approximation of shapes (see [11]) and a suitable choice of such an algorithm would increase the efficiency of the new method in practical applications.

Particular problems that arise if the method is applied to many-fold rotationally symmetric shapes will be discussed in furthcoming papers by the author.

Just to notice that rotationally symmetric shapes appear very often in the industry (as machine made products) but also in the nature (e.g. microorganisms, crystals) and that is a reason for an ongoing research interest ([7,8,10,13,14]).

The method can be applied to open polygonal lines or to the objects that are composed of several polygonal lines. That is of an particular interest when working with incomplete data and with shapes whose boundaries are not extracted completely.

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. V. H. S. Ha, J. M. F. Moura, "Afine-Permutation Invariance of 2-D Shape," *IEEE Transanctions on Image Processing,* Vol. 14. No. 11, pp. 1687-1700, 2005.
4. B.K.P. Horn, *Robot Vision,* MIT Press, Cambridge, MA, 1986.
5. R. Jain, R. Kasturi, B.G. Schunck, *Machine Vision,* McGraw-Hill, New York, 1995.
6. R. Klette, A. Rosenfeld *Digital Geometry,* Morgan Kaufmann, San Francisco, 2004.
7. J.-C. Lin, "Universal Principal Axes: An Easy-to-Construct Tool Useful in Defining Shape Orientations for Almost Every Kind of Shape," *Patt. Rec.,* Vol. 26, pp. 485–493, 1993.
8. J.-C. Lin, "The Family of Universal Axes," *Patt. Rec.,* Vol. 29, pp. 477-485, 1996.
9. R.R. Martin, P.C. Stephenson, "Putting Objects into Boxes," *Computer Aided Design,* Vol.20, pp. 506-514, 1988.
10. V. Shiv Naga Prasad, B. Yegnanarayana, "Finding Axes of Symmetry from Potential Fields," *IEEE Trans. Image Processing,* Vol. 13, No. 12, pp. 1559-1556, 2004.
11. P.L. Rosin, "Techniques for Assessing Polygonal Approximations of Curves," *IEEE Trans. PAMI,* Vol. 19, No. 6, pp. 659-666, 1997.
12. D. Shen, H.H.S. Ip, "Generalized Affine Invariant Normalization," *IEEE Trans. PAMI,* Vol. 19, No. 5, pp. 431-440, 1997.
13. D. Shen, H.H.S. Ip, K.K.T. Cheung, E.K. Teoh, "Symmetry Detection by Generalized Complex (GC) Moments: A Close-Form Solution," *IEEE Trans. PAMI,* Vol. 21, No. 5, pp. 466–476, 1999.
14. W.H. Tsai, S.L. Chou, "Detection of Generalized Principal Axes in Rotationally Symmetric Shapes," *Pattern Recognition,* Vol. 24, pp. 95-104, 1991.
15. J. Žunić, L. Kopanja, J.E. Fieldsend: "Notes on Shape Orientation where the Standard Method Does not Work," *Pattern Recognition*, Vol. 39, No. 5, pp. 856-865, 2005.
16. J. Žunić, P.L. Rosin, L. Kopanja: "On the Orientability of Shapes," *IEEE Transactions on Image Processing*, Vol. 15, No. 11, pp. 3478-3487, 2006.

LPP and LPP Mixtures for Graph Spectral Clustering

Bin Luo and Sibao Chen

Key Lab of Intelligent Computing & Signal Processing of Ministry of Education,
Anhui University, Hefei, Anhui, 230039, China
luobin@ahu.edu.cn, joysbc@163.com

Abstract. In this paper, we concentrate on graph clustering by using graph spectral features. The leading eigenvectors or the spectrum of graphs and derived feature inter-mode adjacency matrix are used. The embedding methods are the Locality Preserving Projection(LPP) and the mixtures of LPP. The experiment results show that although both of the conventional LPP and the LPP mixtures can separate the different graphs into outstanding clusters, the conventional LPP outperforms the LPP mixtures in the sense of compactness for graph clustering.

Keywords: Graph Clustering, Locality Preserving Projection, Mixture models, Graph Spectra.

1 Introduction

Many pattern recognition and computer vision tasks can be charactorised by relational graph analysis and recognition. These include image segmentation, data-base organisation, object recognition and clustering. Although evolved for several decade pattern recognition is powerful enough to handling many problems in practice, it is still difficult to deal with relational structures. The reasons are two-fold. First, graphs are not in nature vectors. While conventional pattern recognition techniques constructs shape-spaces from vectors. It is not straightforward to convert graphs into vectors. Second, In practical, usually there exists structural noise or disturbance, and graphs are in difference size. Hence, graph matching is inexact in nature, and graph clustering faces the difficulty of different dimensional vectors.

Graph similarity and graph distance have attracted enormous research for more than two decades. The idea of using graph edit distance was first explored by Fu and his co-workers [7,17]. Here edit distances are computed using separate costs for the relabeling, the insertion and the removal of both nodes and edges. Recently, Bunke [1] has shown that the graph edit distance and the size of the maximum common subgraph are related under certain restrictions on the edge and node edit costs. Torsello and Hancock [19] have exploited this observation to efficiently compute tree-edit distance. Another approach to computing graph similarity is to adopt a probabilistic framework. Here there are two contributions

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 118–127, 2006.

worth mentioning. First, Christmas, Kittler and Petrou [3] have developed an evidence combining framework for graph-matching which uses probability distribution functions to model the pairwise attribute relations defined on graph-edges. Second, Wilson and Hancock [20] show how to measure graph-similarity using a probability distribution which models the number of relabelling and graph-edit operations when structural errors are present.

A number of vector space embedding techniques can be found in literatures. These include traditional Principle Component Analysis(PCA), Independent Component Analysis(ICA) and Multidimensional Scaling. Locally Linear Embedding method has been published by Roweis and Saul[16]. More recently, He and his coworkers proposed the Locality Preserving Projection(LPP) method [10]. To embed graphs in feature space, Luo, Wilson and Hancock[13] extracted some graph features based on graph spectra. The spectral features are used to embed graphs in the feature space. Several concrete examples of graph clustering applications can be found in literatures. For example, the organisation of large structural data-bases [18] or the discovery of the view-structure of objects [4].

In this paper, we propose a mixtures of LPP based on the conventional LPP and the PCA mixture models. The LPP mixture model is used for graph clustering together with the conventional LPP. The vector features are the graph spectrum and the inter-mode adjacency matrix. Experiments on three model house images are conducted. The Davies-Bouldin index serves for the cluster validation.

2 Spectral Graph Representation

In this paper, we are concerned with a set of graphs $G_1, G_2, .., G_k, ..., G_N$. The kth graph is denoted by $G_k = (V_k, E_k)$, where V_k is the set of nodes and $E_k \subseteq V_k \times V_k$ is the edge-set. Our approach in this paper is a graph-spectral one. For each graph G_k we compute the adjacency matrix A_k. This is a $|V_k| \times |V_k|$ matrix whose element with row index i and column index j is

$$A_k(i,j) = \begin{cases} 1 & \text{if } (i,j) \in E_k \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

From the adjacency matrices $A_k, k = 1...N$ at hand, we can calculate the eigenvalues λ_k by solving the equation $|A_k - \lambda_k I| = 0$ and the associated eigenvectors ϕ_k^ω by solving the system of equations $A_k \phi_k^\omega = \lambda_k^\omega \phi_k^\omega$, where ω is the eigenmode index. We order the eigenvectors according to the decreasing magnitude of the eigenvalues, i.e. $|\lambda_k^1| > |\lambda_k^2| > \ldots |\lambda_k^{|V_k|}|$. The eigenvectors are stacked in order to construct the modal matrix $\Phi_k = (\phi_k^1 | \phi_k^2 | \ldots | \phi_k^{|V_k|})$.

With the eigenvalues and eigenvectors of the adjacency matrix to hand, the spectral decomposition for the adjacency matrix of the graph indexed k is

$$A_k = \sum_{\omega=1}^{|V_k|} \lambda_k^\omega \phi_k^\omega (\phi_k^\omega)^T \tag{2}$$

If $\Lambda_k = diag(\lambda_k^1, ..., \lambda_k^{|V_k|})$ is the diagonal matrix with the eigenvalues of A_k as diagonal elements, then the spectral decomposition of the adjacency matrix can be written as

$$A_k = \Phi_k \Lambda_k \Phi_k^T \tag{3}$$

Associated with the eigenmode with index ω is the adjacency matrix

$$S_k^\omega = \phi_k^\omega (\phi_k^\omega)^T \tag{4}$$

For each graph, we use only the first n eigenmodes of the adjacency matrix. The truncated modal matrix is

$$\Phi_k = (\phi_k^1 | \phi_k^2 | \dots | \phi_k^n). \tag{5}$$

2.1 Leading Eigenvalues

Our first vector of spectral features is constructed from the ordered eigenvalues of the adjacency matrix. For the graph indexed k, the vector is

$$B_k = (\lambda_k^1, \lambda_k^2, ..., \lambda_k^n)^T. \tag{6}$$

This vector represents the spectrum of the graph G_k.

2.2 Inter-mode Adjacency Matrix

The second representation is found by projecting the adjacency matrix onto the basis spanned by the eigenvectors. The projection or inter-mode adjacency matrix is given by

$$U_k = \Phi_k^T A_k \Phi_k \tag{7}$$

The element of the matrix with row index u and column index v is

$$U_k(u, v) = \sum_{i \in V_k} \sum_{j \in V_k} \Phi_k(i, u) \Phi_k(j, v) A_k(i, j) \tag{8}$$

These matrices are converted into long vectors. This is done by stacking the columns of the matrix U_k in eigenvalue order. The resulting vector is $B_k = (U_k(1,1), U_k(1,2), ..., U_k(1,n), U_k(2,1)....., U_k(2,n,), ...U_k(n,n))^T$. Each entry in the long-vector corresponds to a different pair of spectral eigenmodes.

3 Locality Preserving Projection(LPP)

Given a set of n-dimensional training samples $x_i, i = 1, 2, ..., N$, a similarity matrix S is constructed, which can be Gaussian weight or uniform weight of Euclidean distance using k-neighborhood or ε-neighborhood. Considering the problem of mapping a point in n-dimensional (Euclidean) space to a point in d-dimensional space, connected points stay as close together as possible and the intrinsic geometry of the data and local structure is preserved. Let $y_i = w^T x_i$ be the one-dimensional representation of original data vector x_i. A reasonable criterion for choosing this map is to minimize the following objective function [8][9]:

$$\min \sum_{ij} (y_i - y_j)^2 S_{ij}, \tag{9}$$

By simple algebra operation, we see that $\sum_{i<j} S_{ij}(y_i - y_j)^2 = w^T XLX^T w$, where $X = [x_1, x_2, ..., x_N]$ and $L = D - S$ is a Laplacian matrix. D is a diagonal matrix with D_{ii} being column (or row) sum of S, $D_{ii} = \sum_j S_{ij}$. Matrix D provides a natural measure on the vertices of the graph, corresponding to the original images. The bigger the value D_{ii} (corresponding to the ith sample) is, the more "important" is the vertex y_i. Furthermore, to remove an arbitrary scaling factor in the embedding, a constraint is imposed as the following:

$$\sum_i D_{ii} y_i^2 = 1 \Rightarrow w^T XDX^T w = 1. \tag{10}$$

Now the minimization problem is reduced to be:

$$\arg \min_{w^T XDX^T w = 1} w^T XLX^T w. \tag{11}$$

The transformation vector w that minimizes the objective function is given by the minimum generalized eigenvector solution to the generalized eigenvalue problem:

$$XLX^T w = \lambda XDX^T w. \tag{12}$$

Note that the matrices XLX^T and XDX^T are both symmetric and positive semidefinite. And the vectors $w_i (i = 1, 2, ..., d)$ that minimize the objective function are the generalized eigenvectors associated with the d smallest generalized eigenvalues.

4 LPP Mixture Models

In a mixture model, a set of n-dimensional data $x_1, ...x_N$ is partitioned into several clusters. They are assumed to be random observations generated independently from a mixture of M-component probability density function with unknown proportion $\pi_1, ..., \pi_M$

$$f(x; \Theta) = \sum_{j=1}^{M} \pi_j f_j(x; \theta_j), \tag{13}$$

where mixing proportions π_j are nonnegative and sum to one and where $f_j(x; \theta_j)$ denotes the conditional probability density function (p.d.f.) of x belonging to the jth component parameterized by θ_j. Usually these $f_j(x; \theta_j)$ are assumed to be Gaussian density, that is

$$f_j(x; \theta_j) = \frac{1}{(2\pi)^{n/2} |\Sigma_j|^{1/2}} \times \exp\{-\frac{1}{2}(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j)\}, \tag{14}$$

where μ_j and Σ_j are the mean and covariance of the jth component (cluster), respectively.

The number of data required to estimate the parameters of density functions defined on high dimensional spaces increase at least proportionally to the square of the dimensionality, which is curse of dimensionality. So we use the PCA technique here to reduce the dimensionality of the feature space. Let $E(x)$ be the expectation of the random vector x. Then by PCA, a set of n-dimensional data $x_i, i = 1, ..., N$ is reduced to a set of m-dimensional feature data $s_i = T^T(x_i - E(x)), i = 1, ..., N$, where $m \leq n$, $T = (w_1, ..., w_m)$ and w_i is the eigenvector corresponding to the ith largest eigenvalue of the sample covariance $C = (1/N)\sum_{i=1}^{N}(x_i - E(x))(x_i - E(x))^T$. Two properties of PCA are maximizing retained variance and minimizing the squared reconstruction error.

We construct PCA mixture model which combines the above mixture model and PCA technique in a way that the component density of the mixture model can be estimated on the PCA transformed space as

$$\begin{cases} f(x;\Theta) = \sum_{j=1}^{M} \pi_j f_j(x;\theta_j) \\ f_j(x;\theta_j) = f_j(s_j;\theta_j) \end{cases} \tag{15}$$

where $s_j = T_j^T(x - \mu_j)$. Due to the orthogonality of the transform matrix T_j, s_j are decorrelated and have diagonal covariance $\Sigma_j^s = E(s_j s_j^T) = diag(\lambda_{j,1}, ..., \lambda_{j,m})$, where $\lambda_{j,i}$ is the ith largest eigenvalue of the feature covariance matrix Σ_j^s in the jth cluster. So the conditional density $f_j(s_j;\theta_j)$ of the PCA feature vectors in the jth cluster can be simplified as

$$f_j(s_j;\theta_j) = \frac{1}{(2\pi)^{m/2}|\Sigma_j^s|^{1/2}} \times \exp\{-\frac{1}{2}s_j^T \Sigma_j^{s-1} s_j\} \tag{16}$$

$$= \prod_{i=1}^{m} \frac{1}{(2\pi)^{1/2}\lambda_{j,i}^{1/2}} \times \exp\{-\frac{s_{j,i}^2}{2\lambda_{j,i}}\} \tag{17}$$

Here no Gaussian error term is occurred and can be considered as a simplified form of the Tipping and Bishop model [14].

The parameters of PCA mixture model can be estimated by an EM algorithm [6]. E-step and M-step are executed alternately until the likelihood undergoes no further changes. Suppose $\Theta^{(k)}$ is the estimation of Θ obtained after the kth iteration of the algorithm. Then at the $(k+1)$th iteration,
E-step: The posterior probability that x_i belongs to the jth component z_{ij} is computed as

$$\hat{z}_{ij}^{(k)} = \frac{\hat{\pi}_j^{(k)} f_j(x_i, \theta_j^{(k)})}{\sum_{l=1}^{M} \hat{\pi}_l^{(k)} f_l(x_i, \theta_l^{(k)})}. \tag{18}$$

M-step: The mixing proportions π_j are updated as

$$\hat{\pi}_j^{(k+1)} = \sum_{i=1}^{N} \hat{z}_{ij}^{(k)}/N. \tag{19}$$

And the estimates of μ_j and Σ_j are updated as

$$\hat{\mu}_j^{(k+1)} = \frac{\sum_{i=1}^{N} \hat{z}_{ij}^{(k)} x_i}{\sum_{i=1}^{N} \hat{z}_{ij}^{(k)}}, \tag{20}$$

$$\hat{\Sigma}_j^{(k+1)} = \frac{\sum_{i=1}^{N} \hat{z}_{ij}^{(k)} (x_i - \hat{\mu}_j^{(k+1)})(x_i - \hat{\mu}_j^{(k+1)})^T}{\sum_{i=1}^{N} \hat{z}_{ij}^{(k)}}. \tag{21}$$

The new eigenvalue parameters $\lambda_{j,i}$and the new eigenvector (PCA basis) parameters $w_{j,i}$ are obtained by selecting the largest m eigenvalues as

$$\hat{\Sigma}_j^{(k+1)} w_{j,i}^{(k+1)} = \lambda_{j,i}^{(k+1)} w_{j,i}^{(k+1)}, \tag{22}$$

for all $i = 1, ..., m$, $j = 1, ..., M$. Repeat the above two steps until convergence and we will get the parameters of the mixture model.

LPP has only one transformation matrix over all data, which is not enough for the recognition of complex data with many classes and high variations. To improve the performance of LPP, we propose to use LPP mixture model that uses several transformation matrices over all data. PCA mixture model is used to partition the set of all data into an appropriate number of clusters and LPP is applied to each cluster, independently.

After applying PCA mixture model, we obtain several clusters of training samples by posterior probability. For each cluster, we then apply the locality preserving projections algorithm via QR decomposition (LPP/QR)[2]. Note that QR decomposition is more efficient than SVD numerically. It takes QR decomposition of original data matrix and turns to solve generalized eigenvector problem of matrices with $(N_k \times N_k)$ size at most, where N_k is the number of training samples in the kth cluster. This algorithm is especially efficient for under-sampled problem of high dimension data such as images and text data, where the dimension of sample n is greater than the number of training samples N.

5 Cluster Validities

To compare the performances of different clustering methods, researchers have developed many cluster validation methods. These include Davies-Bouldin index[5], Silhouette index[15], Dunn index[11] and C index[12]. The cluster validation indices measure the quality of clusterings. The smaller the index value, the more compact the clusters, the well separated the clusters, and hence the better performance the clustering.

In this paper, we only used traditional k-means clustering. Instead of comparing different clusterings, we compare difference graph features and projection methods by using the clustering validation indices. The validation index values reflect the quality of the graph features and the projection methods.

We adopt Davies-Bouldin index in this paper. Given a set of graph feature vectors $G = g_1, g_2, ..., g_n$ and a k-means clustering of G stored in $C = c_1, c_2, ..., c_M$, where M is the number of clusters, the Davies-Bouldin is defined as follows:

$$DB = \frac{1}{M}\sum_{i=1}^{M} \max_{j=1,2,\ldots,M and j\neq i} d_{ij}, \tag{23}$$

where

$$d_{ij} = \frac{\sigma_i + \sigma_j}{d(c_i, c_j)}, \tag{24}$$

σ_i is the average distance of all the points in cluster i to it's cluster centre c_i in the projected feature space, $d(c_i, c_j)$ is the distance between the two cluster centres c_i and c_j. If the clustering is good, the validation index will be small.

6 Experiments

The aim in this section is to test the proposed LPP mixtures and the LPP projecting methods on graph clustering. The two graph spectral features used are the graph spectrum or the eigenvalues and the inter-mode adjacency matrix of the graphs which were suggested by Luo and Hancock[13]. Two kinds of projecting methods LPP and LPP mixture models are used for the experiments.

The graphs are generated from three model house images. Corner points are extracted as the feature points serving as the nodes of the graphs. The edges of the graphs are generated by using Delauney triangulations on the node set. Examples of the model house images are shown in Figure 1. The first row is the CMU house images, the second row is the INRIA MOVI house images and the last row is the chalet images. The generated graphs are shown in Figure 2 as the same order of Figure 1.

Two sets of experiment are conducted in this paper. The first experiment aims to compare the performance of the two projecting methods LPP and LPP mixtures on the graph spectrum or leading eigenvalues of the graphs.

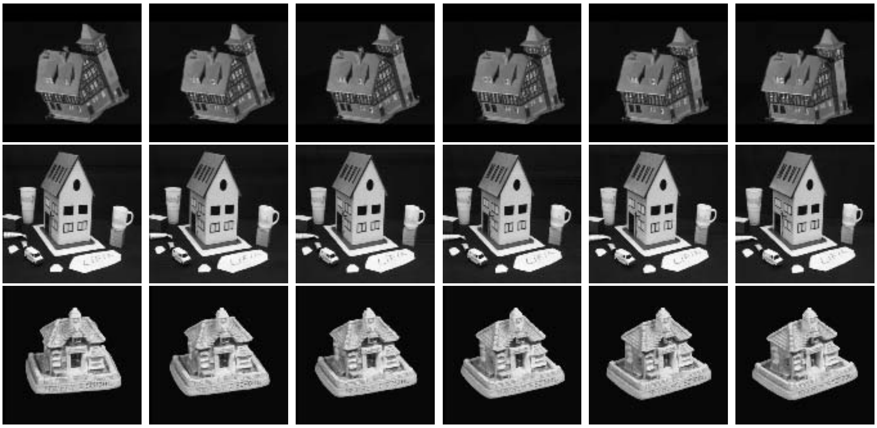


Fig. 1. Test images of the three house sequences. The first row is the CMU sequence, the second row is the MOVI sequence, the last row if the chalet sequence.

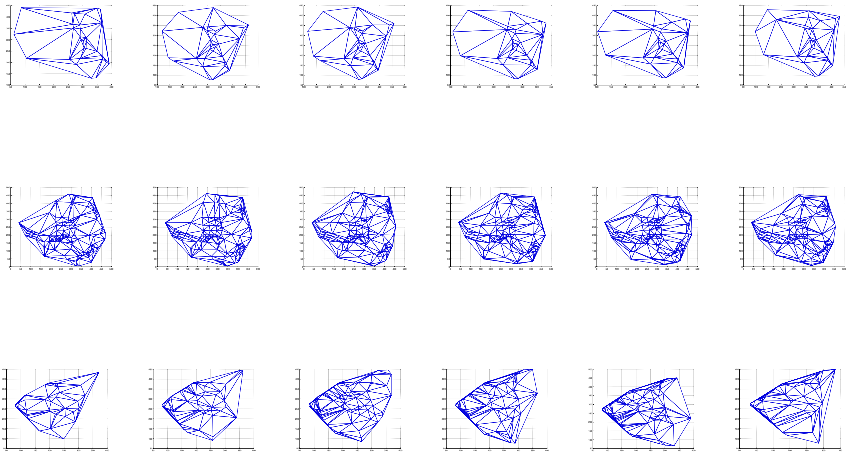


Fig. 2. Graph representation of the three house sequences. The first row is the CMU sequence, the second row is the MOVI sequence, the last row if the chalet sequence.

Eigen-decomposition is conducted on the graph adjacency matrices. The leading eigenvalues are used for graph embedding. The configuration of the projected feature points in 3D space is shown in Figure 3. From the plot we can see that the three types of houses are well separated in both cases, but the compactness and the separation of different clusters of the results from LPP is generally better than the LPP mixtures.

The next experiment uses the spectral feature of the inter-mode adjacency matrix for the embedding. From Figure 4, we can see that both of the embedding are less as compact as the one from the LPP embedded graph spectrum.

To compare the different methods quantitatively, Davies-Bouldin index cluster validation is used to verify the compactness of the resulting clusters. From Table 1 we can see that in both cases of the graph spectrum and inter-mode adjacency matrix features, LPP outperforms LPP mixture for graph clustering in the sense of cluster compactness. Although in some cases the process of LPP is slower than that of the LPP mixture model.

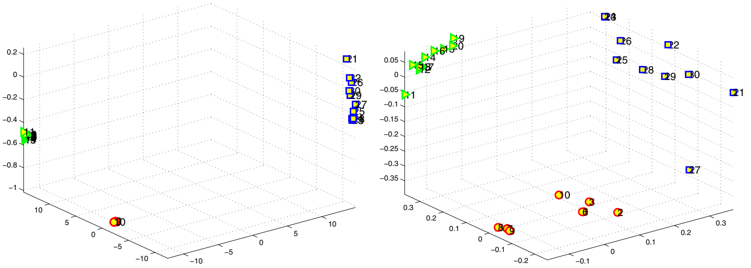


Fig. 3. Graph spectrum feature space embedding (a)LPP (b)LPP mixture

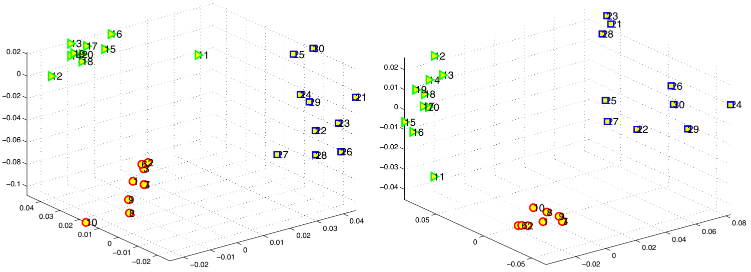


Fig. 4. Inter-mode adjacency matrix feature space embedding (a)LPP (b)LPP mixture

Table 1. Comparison of the LPP amd LPP mixtures for graph clustering

Features	Graph Spectrum		Inter-mode Adjacency Matrix	
Embedding	LPP	LPP Mixtures	LPP	LPP Mixtures
DB	0.4288	0.5172	0.4934	0.8431
Time(s)	3.3750	2.4840	2.6710	2.8120

7 Conclusions

In this paper, we proposed the mixtures of Locality Preserving Projection based on He's LPP model. We aim to use LPP and LPP mixture for graph embedding which is an important issue for structural pattern recognition. We pursue a spectral method of extracting graph spectrum and inter-mode adjacency matrix. The experimental results show that although both of the conventional LPP and the LPP mixtures can separate the different graphs into outstanding clusters, the conventional LPP is outperforms LPP mixtures in the sense of cluster compactness and cluster separations.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No.60375010), and Innovative Research Team of 211 Project in Anhui University.

References

1. H. Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:917–922, 1999.
2. SiBao Chen, HaiFeng Zhao, and Bin Luo. Lpp/qr for under-sampled image recognition. In *IEEE Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, pages 4558–4563, 2005.
3. W.J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE PAMI*, 17(8):749–764, 1995.

4. C.M. Cyr and B.B. Kimia. 3D Object Recognition Using Shape Similarity-Based Aspect Graph. In *ICCV01*, pages I: 254–261, 2001.
5. Davies D. and Bouldin D. A cluster seperation measure. *IEEE T-PAMI*, 1(2):224–227, 1979.
6. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Soc. Ser. B (methodological)*, 39:1–38, 1977.
7. M.A. Eshera and K.S. Fu. An image understanding system using attributed symbolic representation and inexact graph-matching. *Journal of the Association for Computing Machinery*, 8(5):604–618, 1986.
8. X. He and P. Niyogi. Locality preserving projections. In *Proc. Conf. Advances in Neural Information Processing Systems*, 2003.
9. X. He, S.C. Yan, Y. Hu, P. Niyogi, and H.J. Zhang. Face recognition using laplacianfaces. *IEEE T-PAMI*, 27(3):328–340, March 2005.
10. Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and HongJiang Zhang. Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):328–340, 2005.
11. Dunn J. Well seperated clusers and optimal fuzzy partitions. *J. Cybernet*, 4:95–104, 1974.
12. Hubert L. and Schultz J. Quadratic assignment as a general data-analysis strategy. *Br. J. Math. Stat. Psychol.*, 29:190–241, 1976.
13. B. Luo, R.C. Wilson, and E.R. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36(10):2213–2230, October 2003.
14. Tipping M. and Bishop C. Mixtures of probabilistic principal component analysis. *Neural Computating*, 11:443–482, 1999.
15. Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987.
16. S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
17. A. Sanfeliu and K.S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions Systems, Man and Cybernetics*, 13(3):353–362, May 1983.
18. K. Sengupta and K.L. Boyer. Organizing large structural modelbases. *PAMI*, 17(4):321–332, April 1995.
19. Andrea Torsello and Edwin R. Hancock. Efficiently computing weighted tree edit distance using relaxation labeling. *Lecture Notes in Computer Science*, 2134:438–453, 2001.
20. R.C. Wilson and E.R. Hancock. Structural matching by discrete relaxation. *IEEE T-PAMI*, 19(6):634–648, June 1997.

The Invariance Properties of Chromatic Characteristics

Yun-Chung Chung[1], Shyang-Lih Chang[2], Shen Cherng[3], and Sei-Wang Chen[1]

[1] Department of Computer Science and Information Engineering
National Taiwan Normal University, Taiwan
[2] Department of Electronic Engineering, St. John's University, Taiwan
[3] Department of Electrical Engineering, Cheng Shiu University, Taiwan
schen@csie.ntnu.edu.tw

Abstract. An approach to analyzing the degrees of invariance of chromatic characteristics is proposed in this paper. In many vision applications, it is desirable that the chromatic characteristics of objects in images taken under different lighting conditions could remain constant. However, the invariance properties of chromatic characteristics are subject to the lighting conditions. In order to be able to apply to dynamic scenes, we consider three fundamental lighting sources: diffuse, ambient, and directed lightings. Any illumination condition can be approximated as a combination of the three lighting sources. The proposed degree of chromatic invariance is defined based on the chromatic characteristic behaviors under different illumination conditions. A lot of image samples under different illumination conditions are utilized, and from experimental results, we conclude that chromatic characteristics $\{H, C, C_\lambda\}$ are most stable and suitable for the vision applications.

Keywords: Chromatic characteristic invariant, photometric reflectance model, degree of chromatic invariance.

1 Introduction

The colors of an object in images provide lots of information for vision applications, such as object recognition [2], scene interpretation [4], intrinsic images extraction [3], and visual surveillance [9]. The colors in images are determined by the receiving lighting energy in the camera, and actually the intensity of images reflects the brightness of the object that in turn is determined by two essential factors: the amount of light incident on the object and the albedo of the object. It is obviously that even if the same object is pictured, the colors may vary under different illumination conditions.

In many vision applications, it is desirable that the chromatic characteristics of objects could be constant under different lighting conditions. For the purpose to vision applications, we usually want to keep colors of the same object under varying illumination conditions be constant. For example, the license plate recognition algorithm may depend on the colors of the license plate [2], and land mark detection algorithm may rely on the colors of the land marks [4]. In vision applications, the reliable color information under different illumination conditions is important.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 128 – 137, 2006.

Many chromatic characteristics that are invariant to scene geometry and incident illumination have been reported in literature [1, 5, 7]. In this paper, four groups, denoted by {H, C, W, and N} of chromatic characteristics [7] are considered, which can be effectively calculated from the input image. The fundamental invariant chromatic characteristics of the above groups are discussed in later sections. These chromatic characteristics are invariant under some specific imaging conditions. There were five imaging conditions considered in this paper, including uniform illumination, equal energy illumination, colored illumination, matte, dull surfaces, and uniformly colored surfaces. The first three conditions are related to illumination, and the remaining two conditions are associated with object surfaces. We have 11 associated chromatic characteristics from the above imaging conditions, i.e., {H, H_p, C, C_λ, C_p, $C_{\lambda p}$, W, W_λ, $W_{\lambda\lambda}$, N, and N_λ}.

In order to be able to apply to dynamic scenes, instead of directly considering the specific illumination conditions, we turn to three fundamental lighting sources: diffuse, ambient, and directed lightings. Any illumination condition can be approximated as a combination of the three lighting sources. We provide a method to verify the stabilities of the 11 invariant chromatic characteristics under varying illumination conditions. The proposed method calculates the degree of chromatic invariance.

The degree of chromatic invariance is defined based on the chromatic characteristic behaviors under different illumination conditions. In addition, a lot of image samples under different illumination conditions are utilized to verify the degree of chromatic invariance. Finally, from experimental results, we conclude that chromatic characteristics {H, C, C_λ} are most stable and suitable for the vision applications. After first introduction section, we detail the image formation model in Sec. 2, and photometric reflectance model in Sec. 3. Sec. 4 is devoted to the discussion of invariant chromatic characteristics. We then present the experimental results of the degree of chromatic invariance in Sec. 5 and finally give concluding remarks in Sec. 6.

2 Image Formation Model

To illustrate the properties of the chromatic characteristics, we start with an image formation model of a color CCD camera.

$$I^k(p) = T \int_{\lambda} \int_{p \in D_p} q_k(\lambda)\eta(\lambda, d_p)E(\lambda, \boldsymbol{p})d\boldsymbol{p}d\lambda, \quad k = r, g, b, \tag{1}$$

where I^k is the response of the k^{th} camera sensor, p is any image pixel, λ is the light wavelength, T is the exposure time, D_p is the spatial domain of the image pixel p, $\boldsymbol{p}$ is a scene point in D_p, $q_k(\lambda)$ is the spectral sensitivity of the k^{th} camera sensor, $\eta(\lambda, d_p)$ specifies the spectral energy attenuation of the atmosphere, and $E(\lambda, \boldsymbol{p})$ is the amount of spectral energy reflected from scene point $\boldsymbol{p}$.

In the above image formation model, function $q_k(\lambda)$ relating incident spectral energy to camera response is often modeled as $q_k(\lambda) = a_k G(\lambda\text{-}\lambda_k)$, where a_k is a positive constant and $G(\lambda\text{-}\lambda_k)$ is the Gaussian positioned at λ_k. The atmosphere energy attenuation function $\eta(\lambda, d_{\boldsymbol{p}})$ depends on both the light wavelength λ and the distance $d_{\boldsymbol{p}}$ between the scene point $\boldsymbol{p}$ and the camera. According to Allard's law of attenuation, $\eta(\lambda, d_{\boldsymbol{p}}) = e^{-\beta(\lambda)d_{\boldsymbol{p}}} / d_{\boldsymbol{p}}^2$, where $\beta(\lambda)$ is the scattering coefficient. Assuming homogeneous medium, $\beta(\lambda)$ is constant and as a consequence $\eta(\lambda, d_{\boldsymbol{p}})$ is independent of λ, i.e., $\eta(\lambda, d_{\boldsymbol{p}}) \approx \eta(\lambda, d_{\boldsymbol{p}})$, $\forall\lambda$. Furthermore, if the scene's relief is small compared to the average distance from the camera, the scene point $\boldsymbol{p}$'s within the spatial domain D_p of image pixel p can be assumed having similar distances from the camera, i.e., $d_{\boldsymbol{p}} \approx d$, $\forall \boldsymbol{p} \in D_p$, and as a consequence $\eta(d_{\boldsymbol{p}}) \approx \eta$, $\forall \boldsymbol{p} \in D_p$. In reality, the above assumptions seem to be justified. We hence simplify Eq. (1) as

$$I^k(p) = c_k \int_{\lambda} \int_{\boldsymbol{p} \in D_p} E(\lambda, \boldsymbol{p}) G(\lambda - \lambda_k) d\boldsymbol{p} d\lambda, \tag{2}$$

where $c_k = Ta_k\eta$.

3 Photometric Reflectance Model

Many photometric reflectance models [1, 8, 7, 10] have been proposed for describing the spectral energy $E(\lambda, \boldsymbol{p})$. In this study, the model introduced by Geusebroek *et al.* [7] is recruited. The reason we choose the Geusebroek *et al.* model is in view that it generalizes several existing models, including Lambertian reflectance model, Shafer's dichromatic reflectance model [10], and Lambert-Beer transmissive absorption model. The Geusebroek *et al.* model was primarily grounded on the Kubelka-Munk theory [6], which has been shown to be applicable to a wide variety of materials and is well-suited for describing material properties from color measurements. The Kubelka-Munk theory models the reflection and transmission of light in colored layers based on a material dependent scattering and absorption function, through which spectral color formation for both reflecting and transparent materials is integrated into one photometric model. According to Geusebroek *et al.*, the reflected spectral energy $E(\lambda, \boldsymbol{p})$ is described as

$$E(\lambda, \boldsymbol{p}) = i(\lambda, \boldsymbol{p})[(1 - \rho(\boldsymbol{p}))^2 R(\lambda, \boldsymbol{p}) + \rho(\boldsymbol{p})], \tag{3}$$

where $i(\lambda, \boldsymbol{p})$ is the illumination spectral energy, $\rho(\boldsymbol{p})$ is the Fresnel surface reflectance, and $R(\lambda, \boldsymbol{p})$ is the material reflectivity depending on the surface geometry, and the viewing and incidence angles of light. Inevitably, the Geusebroek *et al.* reflectance model is still ideal because of diverse complex scenes. Several assumptions have been incorporated in the Geusebroek *et al.* model, including 1) thick materials, 2) planar surface patches, and 3) uniform colored patches. For outdoor scenes, these assumptions seem to be feasible. In addition to the above assumptions, different imaging conditions were imposed when Geusebroek *et al.* deriving invariant chromatic characteristics based on their proposed model. The imaging conditions considered include equal energy illumination, matte, dull surfaces, uniform illumination, colored illumination, and uniformly colored surfaces.

Accordingly, five groups of invariant chromatic characteristics, denoted by *H*, *C*, *W*, *N*, and *U*, are derived. Each group consists of a fundamental invariant chromatic characteristic and a hierarchy of spectral and spatial derivatives of the fundamental characteristic.

4 Chromatic Characteristics

Many chromatic characteristics that are invariant to scene geometry and incident illumination have been reported in literature [1, 7]. In this paper, four groups, denoted by *H*, *C*, *W* and *N*, of chromatic characteristics are considered, which can be effectively calculated from the input image. The fundamental invariant chromatic characteristics of the five groups are given below.

$$H = \frac{E_\lambda}{E_{\lambda\lambda}},\ C = \frac{E_\lambda}{E},\ W = \frac{E_{\boldsymbol{p}}}{E},\ N = U = \frac{E_{\lambda \boldsymbol{p}}E - E_\lambda E_{\boldsymbol{p}}}{E^2}. \tag{4}$$

They are invariant under different imaging conditions. For example, assuming an equal energy illumination, the spectral components of the light source, $i(\lambda, \boldsymbol{p})$, are constant over the wavelengths, i.e., $i(\lambda, \boldsymbol{p}) = i(\boldsymbol{p})$. Eq. (3) becomes

$$E(\lambda, \boldsymbol{p}) = i(\boldsymbol{p})[(1-\rho(\boldsymbol{p}))^2 R(\lambda, \boldsymbol{p}) + \rho(\boldsymbol{p})]. \tag{5}$$

Differentiating the above equation with respect to λ twice, we obtain

$$E_\lambda(\lambda, \boldsymbol{p}) = i(\boldsymbol{p})(1-\rho(\boldsymbol{p}))^2 R_\lambda(\lambda, \boldsymbol{p}) \text{ and } E_{\lambda\lambda}(\lambda, \boldsymbol{p}) = i(\boldsymbol{p})(1-\rho(\boldsymbol{p}))^2 R_{\lambda\lambda}(\lambda, \boldsymbol{p}). \tag{6}$$

Substituting these equations into $H=E_\lambda/E_{\lambda\lambda}$, we obtain $H = R_\lambda(\lambda, \boldsymbol{p}) / R_{\lambda\lambda}(\lambda, \boldsymbol{p})$. It is clear that *H* depends only on the material reflectivity $R(\lambda, \boldsymbol{p})$. If we repeatedly differentiate *H* with respect to λ and $\boldsymbol{p}$, a hierarchy $H_{\lambda^m p^n}$ of spectral and spatial derivatives of *H* can be obtained. The hierarchy $H_{\lambda^m p^n}$ also depends on $R(\lambda, \boldsymbol{p})$ only.

Let us continue to further assume matte, dull surfaces constituting the scene. Since the Fresnel reflectance coefficients of matte, dull surfaces are close to zero, i.e., $\rho(\boldsymbol{p}) \approx 0$, Eq. (5) is reduced to $E(\lambda, \boldsymbol{p}) = i(\boldsymbol{p})R(\lambda, \boldsymbol{p})$. Differentiating this equation with respect to λ gives rise to $E_\lambda(\lambda, \boldsymbol{p}) = i(\boldsymbol{p})R_\lambda(\lambda, \boldsymbol{p})$. Substituting this equation into $C=E_\lambda/E$, we arrive at $C = R_\lambda(\lambda, \boldsymbol{p}) / R(\lambda, \boldsymbol{p})$ and know that characteristic *C* and its hierarchy depend only on the material reflectivity $R(\lambda, \boldsymbol{p})$ too. The same discussions can be applied to the other groups of chromatic characteristics under different imaging conditions.

Rather than using all the invariant chromatic characteristics suggested by Geusebroek *et al.*, eleven chromatic characteristics, H, H_p, C, C_λ, C_p, $C_{\lambda p}$, W, W_λ, $W_{\lambda\lambda}$, N, and N_λ, are shown for study. Their mathematically formulations are given below and in Eq. (4).

$$\begin{aligned}
&H_p = \sqrt{H_r^2 + H_c^2}\text{ , where } H_i = \frac{E_{\lambda\lambda}E_{\lambda i} - E_{\lambda}E_{\lambda\lambda i}}{E_{\lambda}^2 + E_{\lambda\lambda}^2}\text{ , } i = r,c\text{ ,}\\
&C_{\lambda} = \frac{E_{\lambda\lambda}}{E}\text{ , } C_p = \sqrt{C_r^2 + C_c^2}\text{ , where } C_i = \frac{E_{\lambda i}E - E_{\lambda}E_i}{E^2}\text{ , } i = r,c\text{ ,}\\
&C_{\lambda p} = \sqrt{C_{\lambda r}^2 + C_{\lambda c}^2}\text{ , where } C_{\lambda i} = \frac{E_{\lambda\lambda i}E - E_{\lambda\lambda}E_i}{E^2}\text{ , } i = r,c\text{ ,}\\
&W_{\lambda} = \sqrt{W_{\lambda r}^2 + W_{\lambda c}^2}\text{ , where } W_{\lambda i} = \frac{E_{\lambda i}}{E}\text{ , } i = r,c\text{ ,}\\
&W_{\lambda\lambda} = \sqrt{W_{\lambda\lambda r}^2 + W_{\lambda\lambda c}^2}\text{ , where } W_{\lambda\lambda i} = \frac{E_{\lambda\lambda i}}{E}\text{ , } i = r,c\text{ ,}\\
&N_{\lambda} = \sqrt{N_r^2 + N_c^2}\text{ , where } N_i = \frac{E_{\lambda\lambda i}E^2 - E_{\lambda\lambda}E_iE - 2E_{\lambda i}E_{\lambda}E + 2E_{\lambda}^2E_i}{E^3}\text{ , } i = r,c\cdot
\end{aligned} \tag{7}$$

Note that the hue component of a color is defined as $\tan^{-1}\lambda_{\max}$ where $\lambda_{\max} = -E_{\lambda} / E_{\lambda\lambda}$. The characteristic H expressed as $H=E_{\lambda}/E_{\lambda\lambda}$ is related to the hue and the dominant color of the material. The characteristic H_p, which is the spatial derivative of H, is hence associated with the hue gradient and can be used to detect color edges. The characteristic C defined as $C=E_{\lambda}/E$ stands for normalized color; its spectral derivative C_{λ}, spatial derivative C_p, and spatio-spectral derivative $C_{\lambda p}$ can be use to detect spectral/spatial transitions in object reflectance. The chromatic characteristic W formulated as $W=E_p /E$ indicates intensity normalized edge magnitude. The associated W_{λ} and $W_{\lambda\lambda}$ reflect the spectral slope and curvature of normalized edge magnitude, respectively. Finally, the characteristics N and N_{λ} determine material transitions by detecting changes in object reflectance.

4.1 Measurements

The above chromatic characteristics can be effectively calculated from the input image. Referring to Eqs. (4) and (7), all the formulations of the chromatic characteristics are formed from the terms of E, E_{λ}, $E_{\lambda\lambda}$ and their spatial derivatives. The spatial derivatives of E, E_{λ}, and $E_{\lambda\lambda}$ will easily be obtained by convolving the values of E, E_{λ}, and $E_{\lambda\lambda}$with spatial derivative filters. Therefore, once we determine E, E_{λ} , and $E_{\lambda\lambda}$, all the chromatic characteristics are readily calculated. In the following, we concentrate on how to evaluate E, E_{λ}, and $E_{\lambda\lambda}$ from the input image.

Referring to Eq. (2), this equation states that the response of the k^{th} camera sensor, $I^k(p)$, is obtained by integrating the reflected spectral energy $E(\lambda, \mathbf{p})$ over a certain spatial extend and a certain spectral bandwidth. Along the inverse line of thought, Geusebroek *et al.* introduced a chromatic measurement model characterized by a Gaussian aperture function $G(\lambda;\lambda_k,\sigma_{\lambda})$ to estimate the spectral energy $E(\lambda_k)$ from image intensity $I(\lambda)$, i.e.,

$$E(\lambda_k) = \int_{\lambda} I(\lambda)G(\lambda;\lambda_k,\sigma_{\lambda})d\lambda\cdot \tag{8}$$

The first-order $E_\lambda(\lambda_k)$ and second-order $E_{\lambda\lambda}(\lambda_k)$ spectral derivatives of $E(\lambda_k)$ are then

$$E_\lambda(\lambda_k)=\int_\lambda I(\lambda)G_\lambda(\lambda;\lambda_k,\sigma_\lambda)d\lambda \text{ and } E_{\lambda\lambda}(\lambda_k)=\int_\lambda I(\lambda)G_{\lambda\lambda}(\lambda;\lambda_k,\sigma_\lambda)d\lambda\cdot \tag{9}$$

In discrete cases ($k=r,g,b$), E, E_λ and $E_{\lambda\lambda}$ are actually estimated by linear combinations of given (R, G, B) values. It is also found that E, E_λ and $E_{\lambda\lambda}$ are close to the CIE XYZ basis when taking $\sigma_\lambda=55nm$ for $G(\lambda;\lambda_k,\sigma_\lambda)$. Explicitly,

$$\begin{bmatrix} E \\ E_\lambda \\ E_{\lambda\lambda} \end{bmatrix}=\begin{bmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.35 \\ 0.34 & -0.6 & 0.17 \end{bmatrix}\begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

Based on the above equation, we compute the (E, E_λ, $E_{\lambda\lambda}$) values of any image pixel when given its (R, G, B) values.

5 Experimental Results

In the previous sections, a number of chromatic characteristics were introduced, and the calculation involves a number of uncertainties. First, approximate imaging, photometric reflectance, and chromatic measurement models were employed for calculating chromatic characteristics. Second, high-order spatial and spectral derivatives were involved in the calculation of chromatic characteristics. Third, the invariance properties of chromatic characteristics depend on imaging conditions.

The chromatic characteristics introduced in Sec. 4 include H, H_p, C, C_λ, C_p, $C_{\lambda p}$, W, W_λ, $W_{\lambda\lambda}$, N, and N_λ. Each characteristic can only be invariant under certain imaging conditions. There were five imaging conditions considered in this study, including uniform illumination, equal energy illumination, colored illumination, matte, dull surfaces, and uniformly colored surfaces. The first three conditions are related to illumination, and the remaining two conditions are associated with object surfaces. Since general scenes are considered in this study, the objects constituting a scene can not be known a priori. The imaging conditions concerning object surfaces become impractical and so do the chromatic characteristics, whose invariance properties are subject to the conditions.

Instead of directly considering the three illumination conditions, we turn to three fundamental lighting sources: diffuse, ambient, and directed lightings. Any illumination condition can be approximated as a combination of the three lighting sources. Diffuse lighting comes from the lights refracted from environmental objects. Ambient lighting results from surrounding light sources. Directed lighting comes from a single intense light source. Fig. 1 shows three images of a doll illuminated by a diffuse, an ambient, and a directed light, respectively. The images came from the RVL SPEC-DB database available in the Robot Vision Laboratory at Purdue

University [11]. There are more than 300 color images in the database, which were taken for 100 objects under the three lighting conditions. Objects were made of various kinds of materials. We study the properties of chromatic characteristics using the images provided by the RVL SPEC-DB database.

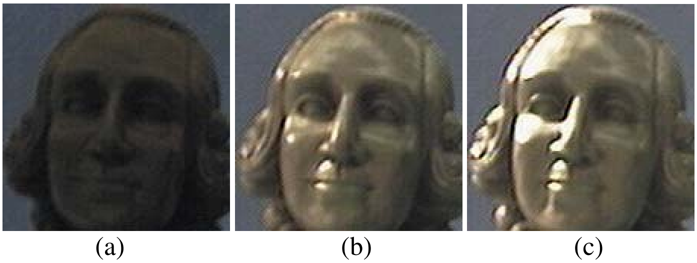

(a) (b) (c)

Fig. 1. A doll under (a) diffuse, (b) ambient, and (c) directed lighting conditions

Let I_1, I_2 and I_3 be the images of the same scene S taken under diffuse, ambient and directed lighting conditions, respectively. Let C specify any chromatic characteristic and $\mathrm{C}_1(\boldsymbol{p})$, $\mathrm{C}_2(\boldsymbol{p})$ and $\mathrm{C}_3(\boldsymbol{p})$ represent the values of the chromatic characteristic at pixel $\boldsymbol{p}$ of the three images, respectively. Ideally, $\mathrm{C}_1(\boldsymbol{p}) = \mathrm{C}_2(\boldsymbol{p}) = \mathrm{C}_3(\boldsymbol{p})$, i.e., the chromatic characteristic of the same material should be invariant under different lighting conditions. Let $\sigma_C(\boldsymbol{p})$ denote the standard deviation of $\mathrm{C}_1(\boldsymbol{p})$, $\mathrm{C}_2(\boldsymbol{p})$ and $\mathrm{C}_3(\boldsymbol{p})$, i.e.,

$$\sigma_C(\boldsymbol{p}) = [\frac{1}{3}\sum_{i=1}^{3}(C_i(\boldsymbol{p}) - \bar{C}(\boldsymbol{p}))^2]^{1/2}, \tag{10}$$

where $\bar{C}(\boldsymbol{p}) = \frac{1}{3}\sum_{i=1}^{3} C_i(\boldsymbol{p})\cdot$

We define the degree of invariance of chromatic characteristic C for scene S as

$$\mathfrak{A}_C(S) = \frac{n_p}{\varepsilon + \sum_{p} \sigma_C(\boldsymbol{p})}, \tag{11}$$

where n_p is the number of image pixels and ε is a small positive number for preventing the denominator from zero.

Figure 2 shows the calculated degrees of invariance of distinct chromatic characteristics. In this figure, the horizontal and the vertical axes represent "scene object" and "logarithm of degree of invariance", respectively. There are eleven curves in the figure, each corresponding to a particular chromatic characteristic. The curves can be roughly divided in terms of variation of degrees of invariance into two groups, one including the curves associated with the characteristics $\{C_p$, $C_{\lambda p}$, W, W_λ, $W_{\lambda\lambda}$, N, $N_\lambda.\}$ and the other including the curves associated with $\{H$, H_p, C, $C_\lambda\}$. The former group has a much larger variation of degrees of invariance than the latter group. A large variation indicates a large instability in degree of invariance with respect to different scene objects. Accordingly, we choose the characteristics $\{H$, H_p, C, $C_\lambda\}$ for further experiments.

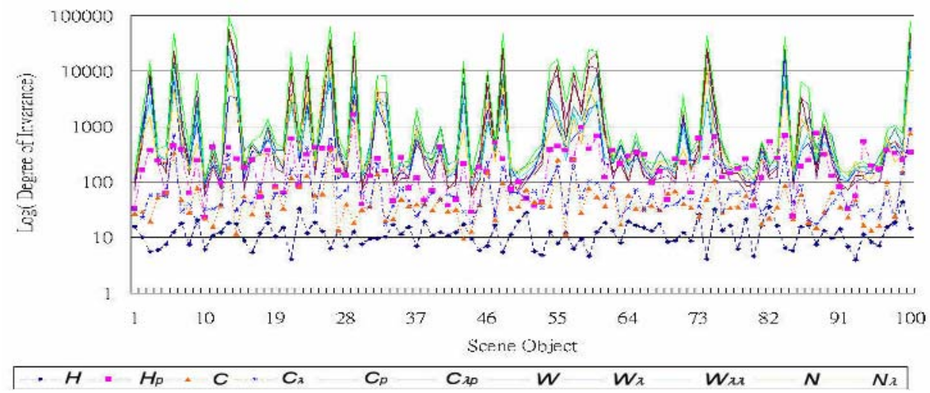
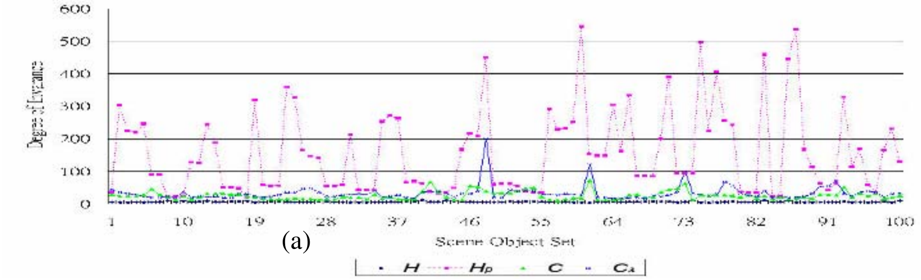


Fig. 2. Degrees of invariance of chromatic characteristics of the same objects under different lighting conditions

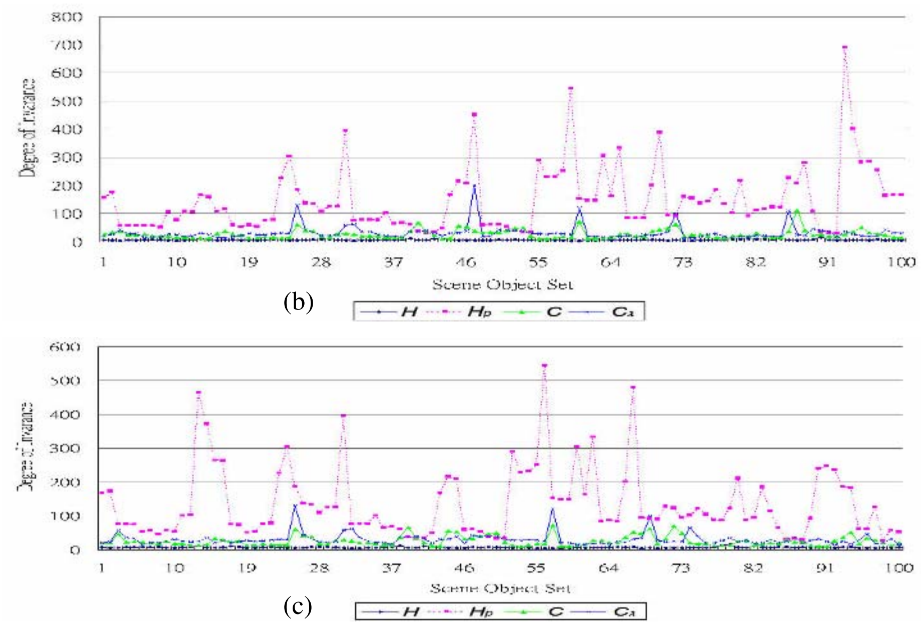
The above experiment investigated the invariance properties of chromatic characteristics with the same objects under different lighting conditions. In the next experiment, we examine the invariance properties of chromatic characteristics with different objects under the same lighting conditions. Let I_1, I_2 and I_3 be the images of three different objects under the same lighting condition L. First of all, we compute the standard deviation, $\sigma_C(\boldsymbol{p})$, of the values of chromatic characteristic C at pixel $\boldsymbol{p}$ of the three images. Next, compute the degree of invariance, $\mathfrak{A}_C(L)$, of chromatic characteristic C by $\mathfrak{A}_C(L) = \dfrac{n_p}{\varepsilon + \sum_p \sigma_C(\boldsymbol{p})}$.

Figures 3 show the calculated $\mathfrak{A}_C(L)$ values of chromatic characteristics {H, H_p, C, C_λ} under diffuse, ambient and directed lighting conditions, respectively. In this figure, the vertical axes represent "degree of invariance" and the horizontal axes represent "set of three distinct objects". There are four curves corresponding to the four chromatic characteristic {H, H_p, C, C_λ}, respectively, in each of the four plots in the figure. In all the plots, the curve associated with the characteristic {H_p} has a larger variation of degree of invariance than those associated with {H, C, C_λ}.



(a)

Fig. 3. Degrees of invariance of chromatic characteristics of distinct scene objects under (a) diffuse lighting condition, (b) ambient lighting condition, and (c) directed lighting condition



(b)

(c)

Fig. 3. (*Continued*)

The similar situation is also observed for the case of distinct scene objects under different lighting conditions (see Fig. 4). We then conclude that chromatic characteristics $\{H, C, C_\lambda\}$ are most stable and suitable for the vision applications.

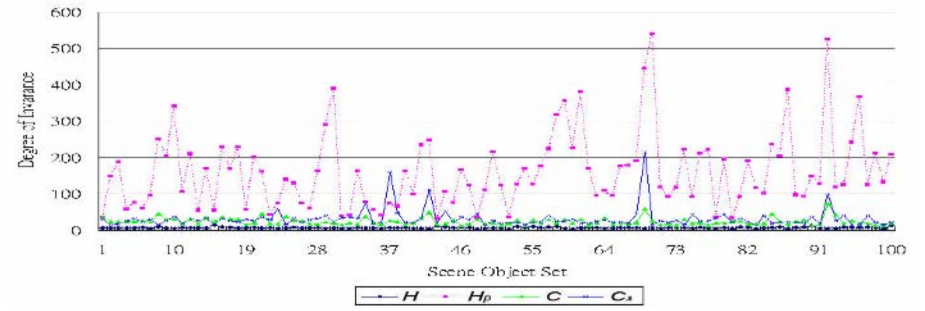


Fig. 4. Degrees of invariance of chromatic characteristics of distinct scene objects under different lighting conditions

6 Concluding Remarks

In this paper, we presented an approach to analyzing the degrees of chromatic invariance. The imaging conditions concerning object surfaces become impractical and so do the chromatic characteristics, whose invariance properties are subject to the conditions. In order to be able to apply to dynamic scenes, we consider three fundamental lighting sources: diffuse, ambient, and directed lightings. Any illumination condition can be approximated as a combination of the three lighting sources. The degree of chromatic invariance is defined based on the chromatic characteristic behaviors under different illumination conditions. The degrees of chromatic invariance are examined with more than 100 sets of image samples. The chromatic characteristics set includes $\{H, H_p, C, C_\lambda, C_p, C_{\lambda p}, W, W_\lambda, W_{\lambda\lambda}, N, \text{and } N_\lambda\}$ is tested, and finally, we conclude that chromatic characteristics $\{H, C, C_\lambda\}$ are most stable and suitable for the vision applications. The procedure of degree of chromatic invariance calculation can be applied to other chromatic characteristic set, and it is suggested to exam the stabilities of the chromatic characteristics before utilizing them into vision applications.

Acknowledgments. This work is supported in part by the National Science Council, Taiwan, Republic of China under contract NSC-94-2213-E-003-008.

References

1. Angelopoulou, E., Lee, S.W., and Bajcsy, R., Spectral Gradient: A Material Descriptor Invariant to Geometry and Incident Illumination, The 7th IEEE Int'l Conf. on Computer Vision, (1999) 861-867.
2. Chang, S.L., Chen, L.S., Chung, Y.C., and Chen, S.W., Automatic license plate recognition, IEEE Trans. on Intelligent Transportation Systems, vol. 5, no. 1, (2004) 42-54.
3. Chung, Y.C., Chang, S.L., Wang, J.M., and Chen, S.W., An Improved Intrinsic Images Extraction from a Single Image with Integrated Measures, IASTED International Conf. on Artificial Intelligence and Applications, Innsbruck, Austria, (2005) 356-361.
4. Fang, C.Y., Chen, S.W., and Fuh, C.S., Automatic Change Detection of Driving Environments in a Vision-Based Driver Assistance System, IEEE Trans. on Neural Networks, vol. 14, no. 3, (2003) 646-657.
5. Finlayson, G.D. and Hordley, S.D., Color Constancy at a Pixel, Journal of the Optical Society of America, 18(2), (2001) 253-264.
6. Geusebroek, J.M., Gevers, T., and Smeulders, A.W.M., The Kubelka-Munk Theory for Color Image Invariant Properties, The 1st Conf. on Color in Graphics, Imaging, and Vision, (2002) 463-467.
7. Geusebroek, J.M., van den Boomgaard, R., Smeulders, A.W.M., and Geerts, H., Color invariance, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 12, (2001) 1338-1350.
8. Healey, G. and Jain, A., Retrieving Multispectral Satellite Images Using Physics-Based Invariant Representations, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18, (1996) 842-848.
9. Kamijo, S., Matsushita, Y., Ikeuchi, K., and Sakauchi, M., Traffic Monitoring and Accident Detection at Intersections, IEEE Trans. on Intelligent Transportation Systems, vol.1, no. 2, (2000) 108-118.
10. Shafer, S.A., Using Color to Separate Reflection Components, Color Resolution Applications, vol. 10, no. 4, (1985) 210-218.
11. The Purdue RVL Specularity Image Database, http://rvl1.ecn.purdue.edu/RVL/ specularity_database/

A Scale Invariant Surface Curvature Estimator

John Rugis[1,2] and Reinhard Klette[1]

[1] CITR, Dep. of Computer Science, The University of Auckland
Auckland, New Zealand
[2] Dep. of Electrical & Computer Engineering, Manukau Institute of Technology
Manukau City, New Zealand
john.rugis@manukau.ac.nz

Abstract. In this paper we introduce a new scale invariant curvature measure, *similarity curvature*. We define a similarity curvature space which consists of the set of all possible similarity curvature values. An estimator for the similarity curvature of digital surface points is developed. Experiments and results applying similarity curvature to synthetic data are also presented.

1 Introduction

There exist numerous well known practical 3D shape applications in computer vision including 3D scan matching, alignment and merging [1], 3D object matching, and 3D object classification and recognition [2]. 3D object databases are an active research area.

It is generally useful to seek *invariant* properties when characterizing 3D objects. At a minimum, translation and rotation invariant characterization are desired as clearly neither of these transformations alters the essential shape property of an object. Surface curvature, a rotation and translation invariant property, meets this requirement [3].

Any characterization that is additionally *scaling invariant* enables determining the equivalence of shapes independent of size. Perhaps not so obvious, practically speaking, this scale invariance would also enable the use of uncalibrated measurement units in 3D digitization (e.g. scanning).

1.1 Curvature

A number of different curvature measures are defined in differential geometry. Curvature is well defined for continuously differentiable lines and surfaces [3]. Planar lines have only a single curvature measure whilst surfaces have a number of curvature measures, all of which are based on *normal curvature*.

On surfaces, two principle curvatures, κ_1 and κ_2, are defined, where κ_1 is the minimum normal curvature and κ_2 is the maximum normal curvature at a given point.

Historically, the *mean curvature* has then been defined as

$$H = (\kappa_1 + \kappa_2)/2$$

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 138–147, 2006.

and the *Gaussian curvature* defined as

$$K = \kappa_1 \kappa_2$$

Additionally, the curvature measure *curvedness* has been defined as

$$C = \sqrt{(\kappa_1^2 + \kappa_2^2)/2}$$

None of the above curvature measures are scale invariant. Also note that, with digital data, there is inherent discontinuity and curvature can only be estimated [4].

1.2 Geometric Invariants

As previously noted, with existing surface curvature definitions we already have translation and rotation invariance. What we now seek is scaling invariance. Shape characterization based on moments has been studied since [5], with varying emphasis on invariance with respect to translation, rotation, reflection, or scaling.

Related work [6], among other things, generalizes and extends the invariance concepts contained in affine differential geometry. Affine invariance is stronger than what we seek in that it includes, for example, squash and stretch transformations.

A number of authors touch on scaling invariant properties in their exploration of multi-scale properties. For example, in [2], firstly surface feature points such as maximum curvature locations are identified. Then triples of feature points are combined using a geometric hashing algorithm in a way that is scaling invariant. Hash tables for various objects of interest are statistically compared to check for similarity matches between different objects.

2 A Scale Invariant Curvature Measure

In this paper, we present a scale invariant curvature measure that can be assigned *at every point on a surface.* We keep in mind that any definition of scale invariant surface curvature must be related to geometric similarity in which it is well known that 1) angles are preserved and 2) ratios of lengths are preserved.

2.1 Similarity Curvature

We begin with some definitions.

Definition 1. *The curvature ratio κ_3 is defined as*

$$\kappa_3 = \frac{\min(|\kappa_1|, |\kappa_2|)}{\max(|\kappa_1|, |\kappa_2|)}$$

In the case when κ_1 and κ_2 are both equal to zero, κ_3 is defined as being equal to zero. Note that $0 \leq \kappa_3 \leq 1$.

Definition 2. *The curvature measure R at a frontier point p is defined as*

$$R(p) = \begin{cases} (\kappa_3, 0) & \text{if signs of } \kappa_1 \text{ and } \kappa_2 \text{ are both positive,} \\ (-\kappa_3, 0) & \text{if signs of } \kappa_1 \text{ and } \kappa_2 \text{ are both negative,} \\ (0, \kappa_3) & \text{if signs of } \kappa_1 \text{ and } \kappa_2 \text{ differ, and } |\kappa_2| \geq |\kappa_1|, \\ (0, -\kappa_3) & \text{if signs of } \kappa_1 \text{ and } \kappa_2 \text{ differ, and } |\kappa_1| > |\kappa_2|. \end{cases}$$

Note that $R(p) \in \mathbb{R}^2$.

We define a *smooth compact 3D set* such that it is compact (i.e., connected, bounded and topologically closed) and curvature is defined at any point of its frontier (i.e., its surface is differentiable at any point).

Theorem 1. *The curvature measure R is (positive) scaling invariant, for any smooth 3D set.*

Proof. Consider a point on a surface and any associated normal curvature κ, as well as the resultant normal curvature κ' after scaling the surface by a factor s. Since, by definition, $\kappa = d\alpha/dl$, and scaling alters length but not angle, we have $\kappa' = \kappa/s$. (Note that the proof could also proceed by considering the effect that scaling has on the osculating circle associated with planar curvature.) Therefore, after scaling, both of the principle curvatures change by the same factor, and the ratio of the principle curvatures is unchanged. Also, neither the signs, nor the relative magnitudes of the principle curvatures are changed by scaling. □

Henceforth, we will refer to the curvature measure R as the *similarity curvature.*

2.2 The Similarity Curvature Space

Since the set of all possible values of the similarity curvature R is a subset of $\mathbb{R}^2$, it is natural to consider a two-dimensional plot representation. Also recall, from differential geometry, that all surface patches on continuous smooth surfaces are locally either elliptic, hyperbolic (saddle-like) or planar.

We introduce the similarity curvature plot template in Figure 1. The horizontal E-axis is for curvature values at locally elliptic surface points. The vertical H-axis is for curvature values at locally hyperbolic surface points. Plotted similarity curvature values will never be off the axes.

Note that the similarity curvature at every point on all spheres from the outside is constant and equal to $(1, 0)$. The similarity curvature on all spheres from the inside is $(-1, 0)$. The similarity curvature on every planar surface, every cylinder and every cone is constant[1] and equal to $(0, 0)$. Note that this is exactly where the Gaussian curvature is equal to zero.

[1] Excluding the cylinder and cone edges and the cone apex.

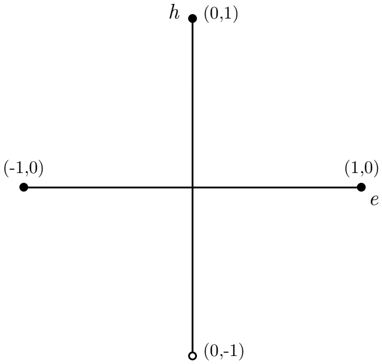


Fig. 1. EH-plot space for similarity curvature

Continuous motion through points on a smooth surface, taking the similarity curvature at each point, results in a related continuous motion in the similarity plot space. We observe, for example, that it is not possible to traverse from similarity curvature (-1,0) to similarity curvature (1,0) without going through similarity curvature (0,0).

However, it is possible to traverse from (0,1) directly to (0,-1). This can be described by saying that the H axis *wraps around.* An example where this wrapping occurs will be given later in this paper.

2.3 Similarity Curvature Estimation

Similarity curvature is estimated using the following process. Firstly the mean and Gaussian curvatures are estimated. The principle curvatures are calculated from the mean and Gaussian curvatures as follows:

$$\kappa_1 = H - \sqrt{H^2 - K}$$

$$\kappa_2 = H + \sqrt{H^2 - K}$$

Then the (estimated) similarity curvature is calculated from the principle curvatures using the definition given earlier in this paper.

The mean and Gaussian curvatures are estimated as done by other authors [7]. Firstly, with reference to the left side of Figure 2, we consider a scan point and, say, six adjacent points. The points are thought to be connected by *edges*, and edges enclose, in this case, six *faces.* We also identify a central angle α_n associated with each face f_n, and each face f_n has area $\mathcal{A}(f_n)$.

The Gaussian curvature is estimated by

$$\tilde{K} = \frac{3(2\pi - \sum \alpha_n)}{\sum \mathcal{A}(f_n)}$$

On the right side of Figure 2, we identify a surface normal vector associated with each face from an edge-on view point. The angle between adjacent face

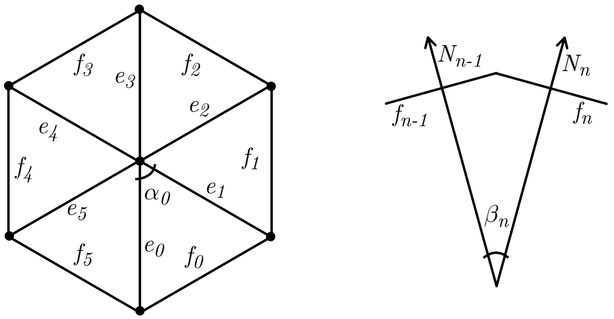


Fig. 2. Curvature estimators: point adjacency on the left and face normals on the right

normals is designated as β. Angle β is positive if the faces form a convex surface (i.e., when viewed from the outside) and β is negative if the faces form a concave surface (i.e., when viewed from the outside).

The mean curvature is estimated by

$$\tilde{H} = \frac{3\sum ||e_n||\,\beta_n}{4\sum \mathcal{A}(f_n)}$$

3 Similarity Curvature Experiments

Synthetic digital data has been created for a number of objects. Each object was digitized by orthogonally scanning from above using a hexagonal grid pattern. The hexagonal scan grid has a *pitch* dimension as shown in Figure 3. Note that, with this scanning method, only the portion of an object that faces towards the scanning source direction gets digitized.

Reference shapes included a sphere, cylinder, ellipsoid and torus. To evaluate the similarity curvature estimation, we considered each reference shape in turn with a scan pitch of 1, then a 10X scaling, and finally a 10X scan resolution. For the 10X scaling, all dimensions and the scan pitch were increased by a factor of 10. For the 10X resolution, the scan pitch was decreased by a factor of 10.

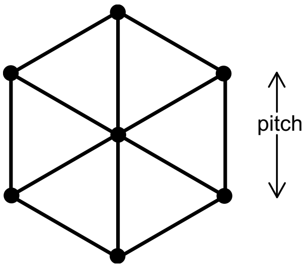


Fig. 3. Scan grid

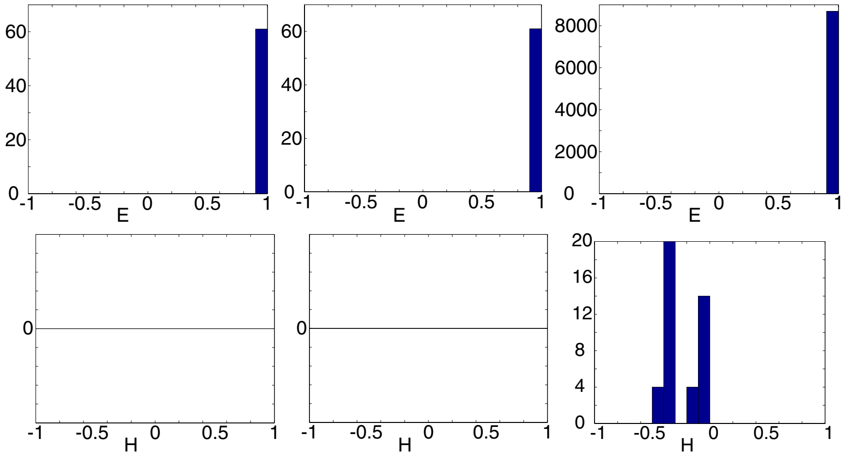


Fig. 4. Sphere EH-histogram: reference, 10x scale, 10x resolution

The resultant similarity curvature values have been accumulated for summary in associated E-axis and H-axis histograms.

Sphere results are shown in Figure 4. The reference sphere has a radius of 5. Observe that the similarity curvature has the constant value of 1 in the E histogram regardless of scale. There are a small number (approximately 0.5 percent) of noise values in the high resolution H histogram.

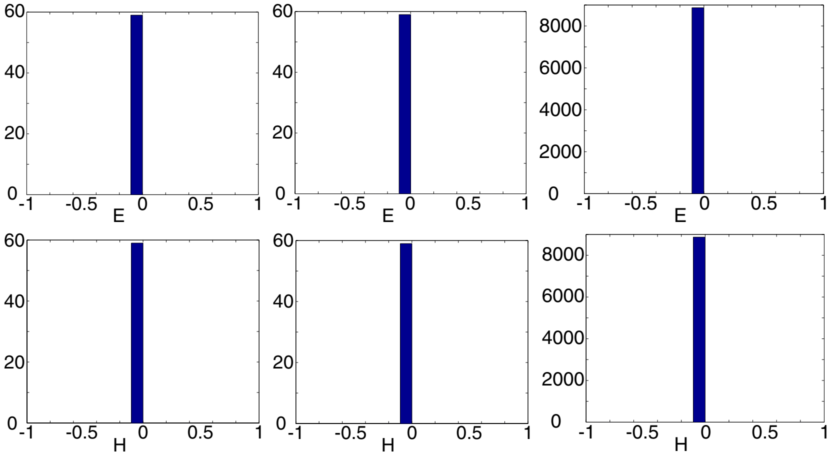


Fig. 5. Cylinder EH-histogram: reference, 10x scale, 10x resolution

Cylinder results are shown in Figure 5. The reference cylinder has a radius of 5 and a height of 4. As expected, in all cases the E and H values are constant at zero.

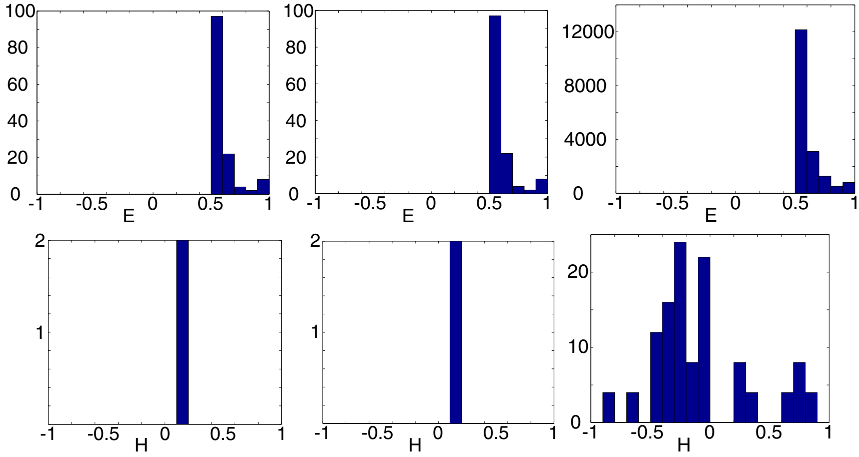


Fig. 6. Ellipsoid EH-histogram: reference, 10x scale, 10x resolution

Ellipsoid results are shown in Figure 6. The reference ellipsoid has axes equal to 6 and 12. As expected, the E values are bounded by 0.5 and 1. Approximately 2 percent noise has accumulated in each of the H histograms.

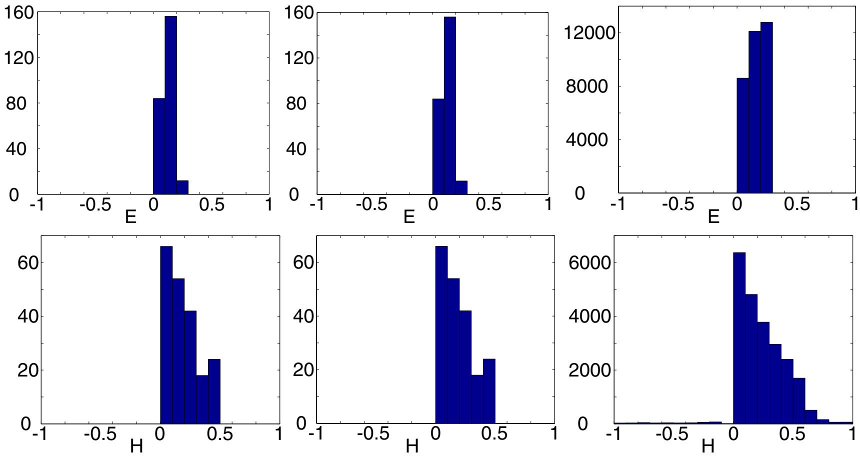


Fig. 7. Torus EH-histogram: reference, 10x scale, 10x resolution

Torus results are shown in Figure 7. The reference torus has an inner radius of 6 and an outer radius of 14. As expected, based on the associated minimum and maximum curvatures, the E values are bounded by 0 and 0.29, and the H values are bounded by 0 and 0.67.

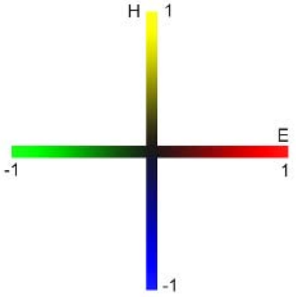


Fig. 8. Shading coded EH-plot axis. (For color see the online version of this LNCS publication.)

3.1 Shading Coded Similarity Curvature

It is possible to assign color coding to similarity curvature values. A color coding of EH-plot axis is shown in Figure 8 where negative E values are green, positive E values are red, negative H values are blue and positive H values are yellow. Shading coded values for similarity curvature can also be used to color each surface point on test objects.

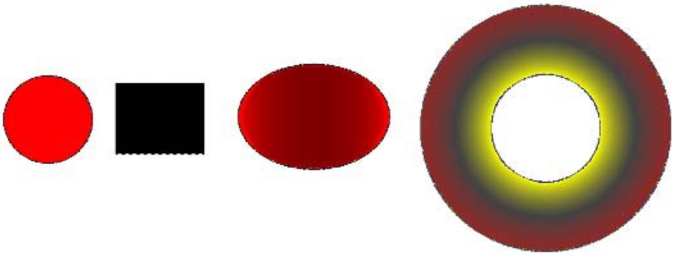


Fig. 9. Shading coded similarity curvature maps: sphere, cylinder, ellipsoid, torus

Shading coded *curvature maps* have been introduced in previous work by the authors [8,9]. A shading coded similarity curvature map for each of the test shapes is shown in Figure 9. The constant curvature of the sphere and the cylinder as well as the smooth transition through curvature values in the curvature maps of the ellipsoid and the torus are readily apparent.

Figure 10 shows the case of a torus having an inner radius of zero. Note the H-axis wrapping near the center of the torus. The bright yellow color transitions change abruptly to bright blue when the H-axis similarity curvature wraps around, changing sign.

3.2 3D Object Detection

Similarity curvature can be used to identify and extract 3D shapes from within complex 3D scan scenes. We may wish to, for example, identify all spheres and spherical patches within a scene no matter what the sphere size or scan resolution.

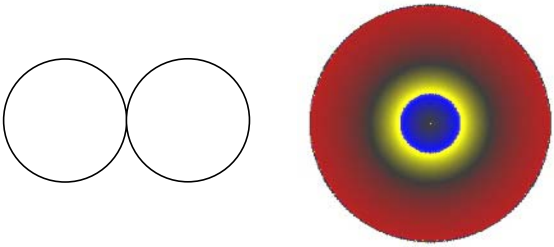


Fig. 10. Torus cross section on left and shading coded similarity curvature

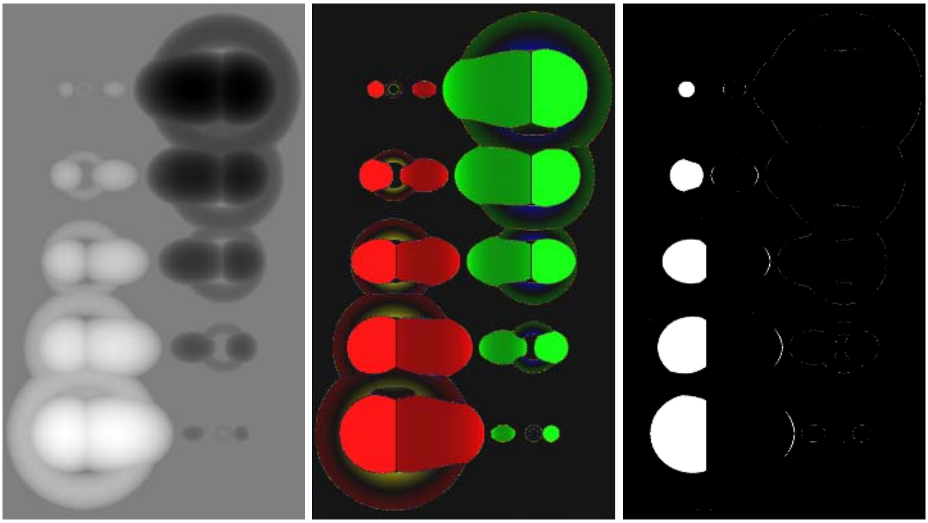


Fig. 11. Test scene depth map, curvature map, and extracted spherical patches

We have constructed a test scan scene containing a surface with five each spherical, ellipsoid and toroid bumps as well as five pits having those same shapes. Some of the pits and bumps overlap. Several representations of the scene are shown in Figure 11. On the left is a shading coded depth map in which points closer to the scan source are white color shaded, and more distant points are shaded black. A color coded similarity curvature map is shown in the middle. Spherical bumps are bright red and spherical pits are bright green. Finally, all of the spherical bump surface patches have been identified and color coded as white in the image on the right.

4 Conclusion

Similarity curvature measure has been defined and an estimator has been introduced and tested. EH-plots as well as a color shaded coding have been presented.

Experiments have demonstrated that similarity curvature can be used to characterize and identify simple synthetic digitized shapes. Further work is anticipated to include applying similarity curvature measure to real world scans and addressing the issue of noisy data.

Acknowledgements. The first author acknowledges financial support from the Department of Electrical and Computer Engineering at Manukau Institute of Technology.

References

1. Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., Fulk, D.: The digital Michelangelo project: 3D scanning of large statues. In: Proc. SIGGRAPH. (2000) 131–144
2. Mokhtarian, F., Bober, M.: *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Sandardization.* Kluwer, Dordrecht (2003)
3. Davies, A., Samuels, P.: *An Introducion to Computational Geometry for Curves and Surfaces.* Oxford University Press, Oxford (1996)
4. Klette, R., Rosenfeld, A.: *Digital Geometry.* Morgan Kaufmann, San Francisco (2004)
5. Hu, M.: Visual problem recognition by moment invariants. IRE Trans. Inform. Theory **8** (1962) 179–187
6. Sapiro, G.: *Geometric Partial Differential Equations and Image Analysis.* Cambridge University Press, Cambridge (2001)
7. Alboul, L., van Damme, R.: Polyhedral metrics in surface reconstruction. In Mullineux, G., ed.: *The Mathematics of Surfaces VI*, Oxford, Clarendon Press (1996) 171–200
8. Rugis, J.: Surface curvature maps and Michelangelo's David. In McCane, B., ed.: Image and Vision Computing New Zealand. (2005) 218–222
9. Rugis, J., Klette, R.: Surface registration markers from range scan data. In Reulke, R., Eckardt, U., Flach, B., Knauer, U., Polthier, K., eds.: Proceedings, Combinatorial Image Analysis: 11th International Workshop, IWCIA. (2006) 430–444

N-Point Hough Transform Derived by Geometric Duality

Yoshihiko Mochizuki[1], Akihiko Torii[2], and Atsushi Imiya[3]

[1] School of Science and Technology, Chiba University 1-33 Yayoi-cho, Inage-ku, 263-8522, Chiba, Japan
[2] Center for Machine Perception, Dept. of Cybernetics, Czech Technical University, Karlovo nám. 13, 121 35 Prague, Czech Republic
[3] IMIT, Chiba University, Yayoi-cho 1-33, Inage-ku, 263-8522, Chiba, Japan

Abstract. We propose an extension of the three-point Randomized Hough transform. Our new Hough transform, which permits a continuous voting space without any cell-tessellation, uses both one-to-one mapping from an image plane to the parameter space and from the parameter space to the image plane. These transforms define a parameter from samples and a line from a parameter, respectively. Furthermore, we describe the classical Hough transform, the randomized Hough transform, the three-point randomized Hough transform and our new Hough transform in a generalized framework using geometric duality.

1 Introduction

In this paper, we propose an extension of the three-point Randomized Hough transform. Our method permits a continuous voting space without any cell-tessellation. Three-point Randomized Hough transform speeds up the computation time and shrinks a search space by selecting randomly three collinear points for the parameter computation in the Randomized Hough transform. There are two possibilities for the selection of three-collinear points. The first method selects three-collinear points for the preprocessing of voting. The second method selects the third point on the line estimated from the first two points as illustrated in Fig. 1 (a). Our method is an extension of the second method, that is, we evaluate the cardinality of sample points on the line estimated from the first two samples as illustrated in Fig. 1(b). In this sense, it is possible to categorize our method into N-points Randomized Hough transform.

The Hough transform, HT in abbreviated form, provides an efficient strategy for the detection of many lines in noisy images [1,2,3,4]. The main idea of the HT is the estimation of parameters of lines by voting. The voting enables us to classify samples in the original image by accumulating the voting and detecting the peaks in the accumulator. Therefore, the detection of the peaks in the accumulator transforms the parameter estimation to search problem. For the detection of the peaks in the accumulator, we traditionally use the accumulator with finite resolution, that is, the accumulator is tessellated to a collection of cells. Therefore, the computation of numbers of accumulation on the cells enables us to detect the peaks of the voting and derive the parameter of lines. The robustness and accuracy of the detected lines depend on the resolution,

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 148–157, 2006.

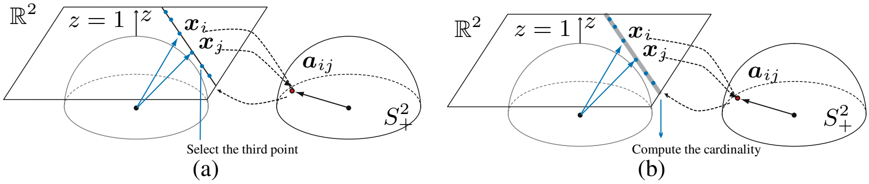


Fig. 1. The three-point randomized Hough transforms and backvoting. (a) The three-point randomized Hough transform with backvoting selects the third point on the line estimated from the first two points. (b) Our method is an extension of (a), that is, we evaluate the cardinality of sample points on the line estimated from the first two samples.

that is, the size of cells. Therefore, the determination of the cell size of the voting space [5,6,7] is a fundamental problem in the derivation of an accurate system for line detection.

The randomized Hough transform, RHT in abbreviated form, is proposed by Oja [8,9] and the three-point Randomized Hough transform is proposed by [10] as an extension of the Randomized Hough transform. The Hough transform is mathematically based on the geometric duality [3,4], which defines one-to-one correspondence between a line on an image plane and a point in the parameter space as illustrated in Fig. 2 (a). Furthermore, conversely geometric duality defines one-to-one inverse correspondence between a line on the parameter space and a point on an image plane as illustrated in Fig. 2 (b). The first mapping derives the Randomized Hough transform, which vote points, and the second inverse mapping defines the classical Hough transform, CHT in abbreviated form, which votes lines. Therefore, the RHT for line detection estimates the parameters using a pair of samples in the original image and a point in the voting space as illustrated in Fig. 3 (a), though the CHT, uses a sample in the original image and a line in the voting space as illustrated in Fig. 3 (b).

In the RHT, the pre-screening process in the sampling phase guarantees the robust estimation of parameters. The three-point RHT is a version of RHT with pre-screening process in the sampling phase [10]. In voting process of the RHT, there exists the voting which are yielded by meaningless selection of samples in the original image, since any two points yield a point in the voting space. For avoiding the meaningless samplings, the three-point RHT tests the collinearity of the selected three points because the samples, which express a line in the original image, must satisfy the collinearity. This preprocessing before voting avoids selecting the meaningless sampling of points in the original image.

Our new Hough transform uses both one-to-one mappings from an image plane to the parameter space and from the parameter space to the image plane as illustrated in Fig 4. The first and second transforms in this process defines a parameter from samples and a line from a parameter, respectively. In the following, we first describe the randomized Hough transform and the three-point randomized Hough transform in our terminology. Second, we define a new randomized Hough transform.

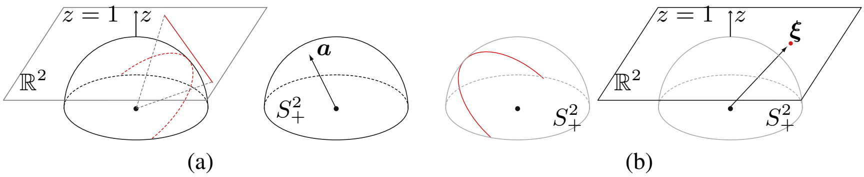


Fig. 2. Geometric duality of a point and a line. (a) If we affix a point in S^2_+, $f(\boldsymbol{a}, \boldsymbol{\xi}) = 0$ defines a line on the plane $z = 1$. (b) If we affix a point in $\mathbb{R}^2$, $f(\boldsymbol{a}, \boldsymbol{\xi}) = 0$ defines a half of a great circle on S^2_+.

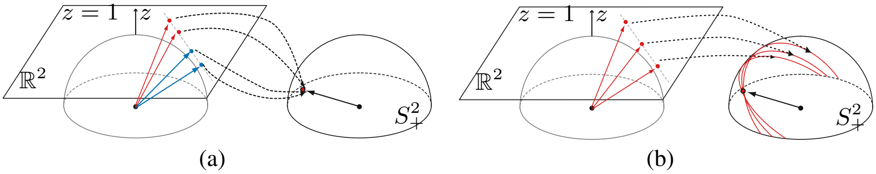


Fig. 3. Hough transform expressed by geometric duality. (a) The randomized Hough transform selects pairs of points on the original image. In the voting space, the randomized Hough transform uses points. (b) The classical Hough transform selects points on the original image. In the voting space, the classical Hough transform uses lines.

2 Hough Transform for Images on a Plane

For $\boldsymbol{\xi} = (x, y, 1)^\top \in \mathbb{R}^3$ and $\boldsymbol{a} = (a, b, c)^\top \in S^2_+$, we define a function

$$f(\boldsymbol{a}, \boldsymbol{\xi}) = \boldsymbol{a}^\top \boldsymbol{\xi}. \tag{1}$$

If we affix $\boldsymbol{a} \in S^2_+$, $f(\boldsymbol{a}, \boldsymbol{\xi}) = 0$ defines a line on the plane $z = 1$ as illustrated in Fig. 2 (a). Conversely, if we affix $\boldsymbol{\xi} \in \mathbb{R}^3$, $f(\boldsymbol{a}, \boldsymbol{\xi}) = 0$ defines a half of a great circle on S^2_+ in Fig. 2 (b). This property is called geometric duality. Hereafter, we call a half of a great circle on S^2_+ as a great circle on S^2_+. Using this geometric duality, we formulate the CHT and RHT, and show these advantages. Since the plane $z = 1$ is topologically equivalent to $\mathbb{R}^2$, we can deal with the lines on the plane $z = 1$ as lines on $\mathbb{R}^2$ when we set $\boldsymbol{\xi} = (\boldsymbol{x}^\top, 1)^\top$ for $\boldsymbol{x} = (x, y)^\top \in \mathbb{R}^2$.

Let the function $u(\tau)$ be

$$u(\tau) = \begin{cases} 1, \text{ if } \tau = 0, \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

Setting $\boldsymbol{P} = \{\boldsymbol{\xi}_i = (x_i, y_i, 1)^\top\}_{i=1}^n$ to be samples in the image plane, $u(\boldsymbol{a}^\top \boldsymbol{\xi}_i) = 1$ defines a plane $\boldsymbol{a}^\top \boldsymbol{\xi}_i = 0$ which passes through the origin of (a, b, c)-space. Since, for $\lambda \neq 0$, $(\lambda a, \lambda b, \lambda c)$ defines the same line, we normalize (a, b, c) as $|\boldsymbol{a}| = 1$, $c > 0$ on S^2_+. A system of equations,

$$\boldsymbol{a}^\top \boldsymbol{\xi}_i = 0, \ \ |\boldsymbol{a}| = 1 \tag{3}$$

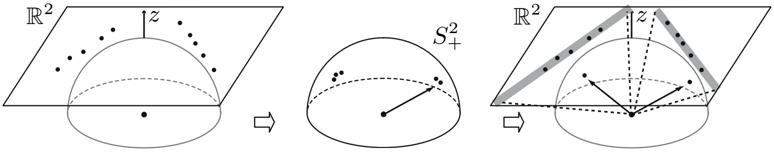


Fig. 4. The randomized Hough transform with a continuous voting space. Our randomized Hough transform uses both one-to-one mappings from an image plane to the parameter space and from the parameter space to the image plane.

expresses the common curve with a plane, which passes through the origin, and the sphere. This geometrical property is called duality of a line and a point on S^2.

For a given point $\boldsymbol{\xi}_i$, the solution of the system of equations,

$$u(\boldsymbol{a}^\top \boldsymbol{\xi}_i) = 1, \ \ \boldsymbol{a} \in S^2_+, \tag{4}$$

defines a great circle on S^2_+, which is a line on the positive constant-curvature manifold. Furthermore, the solution of the system of equations,

$$u(\boldsymbol{a}^\top \boldsymbol{\xi}_i) = 1, \ \ u(\boldsymbol{a}^\top \boldsymbol{\xi}_j) = 1, \tag{5}$$

that is equivalent to the system of equations,

$$\boldsymbol{a}^\top \boldsymbol{\xi}_i = 0, \ \ \boldsymbol{a}^\top \boldsymbol{\xi}_j = 0, \tag{6}$$

where $\boldsymbol{a} = (a, b, c)^\top \in S^2_+$, defines a point as the common point of a pair of the great circles on S^2_+. This geometrical property defines the transform from (x, y)-plane to a point on S^2_+, for

$$\boldsymbol{a}_{ij} = \lambda \frac{\boldsymbol{\xi}_i \times \boldsymbol{\xi}_j}{|\boldsymbol{\xi}_i \times \boldsymbol{\xi}_j|}. \quad \lambda = \pm 1, \tag{7}$$

where λ is selected so that $\boldsymbol{a}_{ij} \in S^2_+$. These mathematical properties derive the voting for RHT expressed by

$$g(\boldsymbol{a}) = \sum_{i \neq j} u(\boldsymbol{a}^\top \boldsymbol{a}_{ij}), \quad \boldsymbol{a} \in S^2_+ \tag{8}$$

as a transform from S^2_+ to $\mathbb{Z}$. These processes are a mathematical expression of point-to-point voting. This procedure provides the RHT for the detection of many lines.

Algorithm 1. *RHT for Line Detection*

```
1  For $\boldsymbol{P} = \{\boldsymbol{x}_i = (x_i, y_i)\}_{i=1}^n$
2  Compute Eq. (7) for randomly selected $\boldsymbol{x}_i$ and $\boldsymbol{x}_j \in \boldsymbol{P}$. Repeat this step for
   N-times where N is a given constant.
3  Compute Eq. (8) as the results of Step 2.
4  Detect $\boldsymbol{a} \in S^2_+$ such that $g(\boldsymbol{a}) >$ T for a given real constant T.
```

In Eqs. (7) and (8), elimination on the meaningless sampling and voting enables us to detect robustly the lines. For the robust computation of parameter $\boldsymbol{a}$, we modify Eq. (8) to the function,

$$g(\boldsymbol{a}) = \sum_{i \neq j \neq k} u(\boldsymbol{a}^\top \boldsymbol{a}_{ijk}), \quad \boldsymbol{a} \in S^2_+ \tag{9}$$

where

$$\boldsymbol{a}_{ijk} = \frac{1}{3}(\boldsymbol{a}_{ij} + \boldsymbol{a}_{jk} + \boldsymbol{a}_{ki}), \tag{10}$$

for three sample points on $\mathbb{R}^2$. Furthermore, Steps 2 and 3 in **Algorithm 1** is modified as follows.

Sub Algorithm 1-1. *Three-Point RHT*

```
2'-1  Repeat Step 2'-2 and Step 2'-3 for N-times where N is a given constant.
2'-2  Select three point $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ randomly.
2'-3  If $|(\boldsymbol{\xi}_i \times \boldsymbol{\xi}_j)^\top \boldsymbol{\xi}_k| < \delta$, then compute $\boldsymbol{a}_{ij}$, $\boldsymbol{a}_{jk}$ and $\boldsymbol{a}_{ki}$ on Eq. (7) and output
      $\boldsymbol{a}_{ijk}$ on (10). Else go to Step 2"-1.
3'    Compute Eq. (9)
```

These modified steps select a triplet of collinear points which may lie on the same line. The RHT with step 2' is called three-point RHT [10].

On the basis of the geometric duality, it is possible to draw a line on the original image plane using a point on the voting space such that, for $\boldsymbol{a} \in S^2_+$ and $\boldsymbol{\xi} = (x, y, 1)^\top$,

$$\boldsymbol{a}^\top \boldsymbol{\xi} = 0. \tag{11}$$

This process is called the back-voting. For the selection of the third point, it is possible to use the back-voting operation. In practical applications, we select a point in string

$$\frac{|\boldsymbol{a}^\top \boldsymbol{\xi}|}{\sqrt{a^2 + b^2}} \leq \delta, \tag{12}$$

for $\boldsymbol{a} \in S^2_+$, $\boldsymbol{\xi} = (x, y, 1)^\top$ and a fixed parameter δ. By adapting the back-voting operation, Step 2' in **Sub Algorithm 1-1** is modified as follows.

Sub Algorithm 1-2. *Back-Voting Three-Point RHT*

```
2"-1  Repeat Step 2"-2 to 2"-4 for N-times where N is a given constant.
2"-2  Compute Eq. (7) for randomly selected $\boldsymbol{x}_i$ and $\boldsymbol{x}_j \in \boldsymbol{P}$.
2"-3  Select $\boldsymbol{x}_k$ such that $\boldsymbol{x}_k \neq \boldsymbol{x}_i$ and $\boldsymbol{x}_k \neq \boldsymbol{x}_j$.
2"-4  If $\frac{|\boldsymbol{a}^\top \boldsymbol{\xi}_k|}{\sqrt{a^2+b^2}} \leq \delta$, then compute $\boldsymbol{a}_{jk}$ and $\boldsymbol{a}_{ki}$, output $\boldsymbol{a}_{ijk}$ on (10), and go to Step
      2"-1. Else repeat going to Step 2"-3 for N'-times where N' is a given constant.
```

Three-point RHT selects a triplet of points which may lie on a line. However, the three-point RHT with the back-voting selects the third point, which passes through a line yielded by the first pair of points. This additional operation enables to shrink the search space for the third point.

3 RHT with a Continuous Voting Space

For samples $\boldsymbol{P} = \{(x_i, y_i, 1)^\top\}_{i=1}^n$ in the image plane, the RHT with the tessellated-cell accumulator is constructed by three procedures as follows.

(a) Sampling and voting based on Eq. (7), that is, transform from a pair of points in the original image to a point on S_+^2.

(b) Accumulation of the voting to tessellated cells based on Eq. (8).

(c) The peak detection in the accumulator.

For the construction of RHT with the continuous voting space, we employ the three procedures as follows.

(A) Sampling and voting based on Eq. (7), that is, transform from a pair of points in the original image to a point on S_+^2.

(B) Back-voting and computation of the cardinality of sample points in the back-voted strip for the estimation of the reliable voting point.

(C) Selection of the reliable voting points on S_+^2 for the determination of lines in the original image.

On the RHT with the continuous voting space, first, we operate the sampling and voting in (A) in the same way of the RHT with the tessellated-cell accumulator in (a). In this procedure, the voting space S_+^2 is continuous. Second, instead of the accumulation in (b), we operate the back-voting and the computation of the cardinality of sample points in the back-voted strip in (B). On the basis of Eq. (11), it is possible to draw a line on the original image plane using a voting point on S_+^2 computed by Eq. (7). For the line drawn by the back-voting, we generate a strip expressed by Eq. (12) as illustrated in Fig 4. Using this strip, we compute the numbers of points, which lie on the strip, in the original image. We express the computation of cardinality as follows.

$$C(\boldsymbol{P}, \boldsymbol{a}, \delta) = |\boldsymbol{P} \cap \{\boldsymbol{\xi}| \ \frac{|\boldsymbol{a}^\top \boldsymbol{\xi}|}{\sqrt{a^2 + b^2}} \leq \delta\}|. \tag{13}$$

This cardinality of sample points in the back-voted strip enables us to express the reliability of the parameter of the line generated from a pair of points. If this cardinality is large, the line may exist as a line in the original image. Conversely, if the cardinality of sample points in the back-voted strip is less than three, the line may not exist as a line in the original image. Finally, instead of the peak detection in (c), we select the points on S_+^2, that satisfy the relation, $C(\boldsymbol{P}, \boldsymbol{a}, \delta) > \overline{\mathrm{T}}$, for a given real constant $\overline{\mathrm{T}}$.

If there do not exist noises in the sample points and the selection of the meaningless sampling of pairs of points, δ in Eq. (13) satisfies $\delta = 0$. Furthermore, for each line, the points $\boldsymbol{a}$ in S_+^2 become identical. Therefore, the cardinality of sample points in the back-voted strip yielded by the points $\boldsymbol{a}$ in S_+^2, which express the same great circle, have the same number.

Since we assume that the sample points in the original image have noise, a line in the original image might be expressed as collection of points in the voting space. However, it is possible to determine a point from the collection of points as the parameter of a line using the cardinality in Eq. (13). The point, which has the largest cardinality, in the voting space is the most reliable as parameter of a line in the original image. On the basis of this reliability, we determine the parameter of lines step-by-step. First, we compute Eq. (13) for all points in the voting space. Second, we select the point, which has the largest cardinality, as the parameter of a line. Third, we remove the collection of points, which lie on the strip defined by the selected point in the voting space, in the original image. This operation merges the collection of points, which may express the same line, in the voting space. Finally, we repeat these operations for the rest of points in the voting space and collection of points in the original image.

These properties based on the geometric duality provide the RHT with the continuous voting space for line detection.

Algorithm 2. *RHT with Continuous Voting Space*

```
1   For $\boldsymbol{P} = \{\boldsymbol{x}_i = (x_i, y_i)\}_{i=1}^{n}$
2   Compute Eq. (7) for randomly selected $\boldsymbol{x}_i$ and $\boldsymbol{x}_j \in \boldsymbol{P}$. Repeat this step for
    N-times where N is a given constant.
3   Compute Eq. (13) for all points $\boldsymbol{a}_{ij}$ on $S^2_+$.
4   Select $\boldsymbol{a}_{\max} \in S^2_+$ such that $\max(C(\boldsymbol{P}, \boldsymbol{a}, \delta))$.
5   If $C(\boldsymbol{P}, \boldsymbol{a}_{\max}, \delta) > \overline{\mathrm{T}}$ for a given real constant $\overline{\mathrm{T}}$, then output $\boldsymbol{a}_{\max}$, remove
    $\boldsymbol{x}_k$ from $\boldsymbol{P}$ which lie on the strip yielded by $\boldsymbol{a}_{\max}$ and go to Step 2. Else exit.
```

The back-voting three-point RHT in **Sub-Algorithm 1-2** selects the third point, which passes through a line yielded by the first pair of points, and avoids the selection of the meaningless sampling of points in the original image. We extend this idea on the RHT with a continuous voting space. For a point $\boldsymbol{a}_{ij}$ computed by Eq. (7), we operate the back-voting and generate the strip in the original image. If the cardinality of sample points in this back-voted strip is larger than three, the point $\boldsymbol{a}_{ij}$ is voted to the voting space S^2_+. Therefore, for a point $\boldsymbol{a}_{ij}$ computed by Eq. (7), this procedure operates the back-voting and computation of the cardinality of sample points in the back-voted strip before the determination of the voting to the voting space S^2_+. For adapting this back-voting before voting Step 2 in **Algorithm 2** is rewritten as follows,

Sub Algorithm 2-1. *Back-Voting Before Voting*

```
2'-1  Repeat Step 2'-2 and Step 2'-4 for N-times where N is a given constant.
2'-2  Compute Eq. (7) for randomly selected $\boldsymbol{x}_i$ and $\boldsymbol{x}_j \in \boldsymbol{P}$.
2'-3  Compute Eq. (13) for $\boldsymbol{a}_{ij}$.
2'-4  If $C(\boldsymbol{\xi}; \boldsymbol{a}_{ij}) > 3$, then vote $\boldsymbol{a}_{ij}$ on $S^2_+$.
```

We summarize the RHT with a continuous voting space for the detection of many lines as follows.

Algorithm 3. *Back-Voting RHT with Continuous Voting Space*

```
1    For $\boldsymbol{P} = \{\boldsymbol{x}_i = (x_i, y_i)\}_{i=1}^{n}$
2-1  Repeat Step 2'-2 and Step 2'-4 an appropriate time.
2-2  Compute Eq. (7) for randomly selected $\boldsymbol{x}_i$ and $\boldsymbol{x}_j \in \boldsymbol{P}$.
2-3  Compute Eq. (13) for $\boldsymbol{a}_{ij}$.
2-4  If $C(\boldsymbol{\xi}; \boldsymbol{a}_{ij}) > 3$, then vote $\boldsymbol{a}$ on $S^2_+$.
3    Compute Eq. (13) for all points $\boldsymbol{a}_{ij}$ on $S^2_+$.
4    Select $\boldsymbol{a}_{\max} \in S^2_+$ such that $\max(C(\boldsymbol{\xi}; \boldsymbol{a}))$.
5    If $C(\boldsymbol{\xi}; \boldsymbol{a}) > \overline{\mathrm{T}}$ for a given real constant $\overline{\mathrm{T}}$, then output $\boldsymbol{a}_{\max}$, remove $\boldsymbol{x}_k$ from
     $\boldsymbol{P}$ which lie on the strip yielded by $\boldsymbol{a}_{\max}$ and go to Step 2. Else exit.
```

Setting $\hat{\boldsymbol{\xi}}_i = \boldsymbol{\xi}_i + \varepsilon$ and $\hat{\boldsymbol{\xi}}_j = \boldsymbol{\xi}_j + \varepsilon$ to be the points on a sphere that includes noise ε, the normal vector is computed as

$$\hat{\boldsymbol{a}}_{ij} = \lambda \frac{\hat{\boldsymbol{\xi}}_i \times \hat{\boldsymbol{\xi}}_j}{|\hat{\boldsymbol{\xi}}_i \times \hat{\boldsymbol{\xi}}_j|} = \boldsymbol{a}_{ij} + \varepsilon \bar{\boldsymbol{a}}_{ij} + O(\varepsilon^2). \tag{14}$$

If $\hat{\boldsymbol{a}}_{ij}$ is voted and accumulated to the voting space with finite resolution, the noise described in Eq. (14) definitively affects the result of estimation of the normal vector through the voting procedure. The RHT with a continuous voting space employs S^2_+ as the voting space. Therefore, in the sense of numerical computation, it is possible to compute the voting point $\boldsymbol{a}_{ij}$ accurately.

Furthermore, the robustness of the RHT with continuous voting space depends on δ, that is, the width of the strip defined for the computation of cardinality. The robustness of the RHT the tessellated-cell accumulator depends on the sizes of cells. Mathematically, the evaluation of robustness in the RHT with continuous voting space is simple compared to the evaluation in the RHT with finite resolution, since it is difficult to tessellate S^2_+ to equi-areal cells as voting cells.

4 Numerical Examples

Fig. 5 shows the result of numerical experiment for line detection using the Hough transform of our new randomized and classical methods. For the experiment, we prepared "A House" as shown in Fig. 5 (a). For the preprocessing of line detection, edge-points are extracted by Canny operator [11,12]. There are 1109 points extracted by Canny operator as shown in Fig. 5 (b). Our randomized Hough transform with a continuous voting space detected 10 lines from the input 1109 points as shown in Fig. 5 (c). For the numerical evaluation of robustness and accuracy of our result, we applied `cvHoughLines2` function, which uses the classical Hough transform, in openCV library [12] to the same edge images. Fig. 5 (d) shows the result detected by the `cvHoughLines2` function.

In the practical experiment, the robustness and accuracy of line detection depend on the parameters in the algorithms. Our randomized Hough transform uses the parameters δ_1, δ_2 and $\overline{\mathrm{T}}$. The parameter δ_1 is the size of strip in **Sub Algorithm 2-1**. Theoretically, the parameter δ_1 is set as 0 since at least three points should pass through an estimated

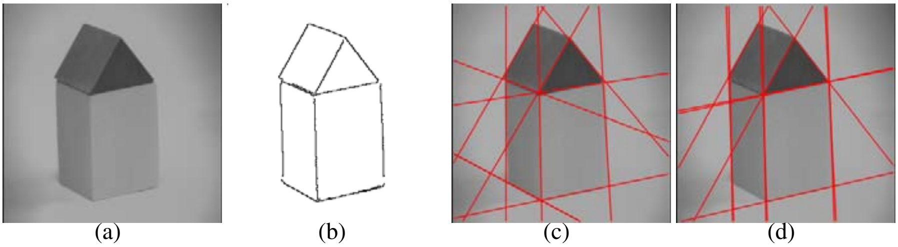


Fig. 5. Numerical examples for "A House". (a) Original image. (b) Edge image extracted by Canny operator. (c) The result using our randomized Hough transform with a continuous voting space. (d) The result using **cvHoughLines2** function in openCV. In (c), we can observe that 10 lines are successfully detected by our method. In (d), some lines are over-detected by the classical Hough transform, since it is difficult and sensitive to determine the parameters of the cell size and threshold of peak detection.

line exactly. However, practically, the points in the digital images are expressed as pixels. This geometric propertiy of points in the digital image implies that the edge-points in the digital image are expressed in the integer coordinates. The line estimated from two integer points may not pass through the third point, which is also expressed as integer, in the digital image, even though these three points express a line in the Euclidean space. Therefore, we set the parameter δ_1 as 0.5 pixel units. The parameter δ_2 is the size of strip in **Algorithm 2**. We set the parameter δ_2 as 2 pixel units. Theoretically, it is possible to define this parameter δ_2 if we know the variance of points to the line. However, it is impossible to compute the variance before the estimation of the line itself. The parameter $\overline{\mathrm{T}}$ is the minimum number of collections of points, which express a line in the image. We set the parameter $\overline{\mathrm{T}}$ as 30 in Fig. 5 since we assumed that the minimum numbers of collection of points should consist of at least 3 % input edge-points. On the other hand, in `cvHoughLines2` function which uses the classical Hough transform, we need to analyze mathematical and practical relations among the sizes of cells in the accumulator and the thresholds of the peak detection in the accumulator. Therefore, it is difficult to define the parameters of the size of cells and the threshold. In this experiments, the selections of the parameters for `cvHoughLines2` function are solved ad-hoc.

In our randomized Hough transform, the geometric properties of parameters are clearly well-defined since our algorithm is constructed based on the geometric duality. Therefore, it is possible to define the appropriate parameters theoretically and practically. Furthermore, for the line detection from edge-points in Fig. 5 (b), there exists two kinds of noises. One is yielded in digitization process and included in each line element. The other is the outliers to each line. The difference in Figs. 5 (c) and (d) can promise that our new randomized Hough transform is robust against both two kinds of noises. Therefore, our new randomized Hough transform detects sufficient numbers of lines from real images compared to the results by `cvHoughLines2` function.

5 Conclusions

We proposed new randomized Hough transform, which permits a continuous voting space without any cell-tessellation. Furthermore, using geometric duality. we described the classical Hough transform, the randomized Hough transform, the three-point randomized Hough transform and our new Hough transform in a generalized framework. In this generalized framework, we geometrically clarified the voting, accumulation, peak-detection and back-voting. Moreover, in the numerical experiments, we showed that our new randomized Hough transform detects lines in images robustly and accurately compared to the traditional method.

References

1. H. Kalviainen, P. Hirvonen, L. Xu and E. Oja. Probabilistic and nonprobabilistic hough transforms: Overview and comparisons. *IVC*, vol. 13, no. 4, pp. 239–252, May 1995.
2. E. Aguado, A.S. Montiel and M.S. Nixon. On the intimate relationship between the principle of duality and the hough transform. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 456, pp. 503–526, 2000.
3. A. Bhattacharya, P. Rosenfeld and I Weiss. Geometric and algebraic properties of point-to-line mappings. *Pattern Recognition*, vol. 36, pp. 483–503, 2003.
4. A. Bhattacharya, P. Rosenfeld and I. Weiss. Point-to-line mappings as hough transforms. *Pattern Recognition Letters*, vol. 3, pp. 1705–1710, 2002.
5. J. Immerkaer. Some remarks on the straight line hough transform. *PRL*, vol. 19, no. 12, pp. 1133–1135, October 1998.
6. N. Kiryati and A.M. Bruckstein. Antiariasing of the hough transform. *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, vol. 53, pp. 213–222, 1991.
7. A. Goldenshluger and A. Zeevi. The hough transform estimator. *The Annals of Statistics*, vol. 32, no. 5, pp. 1908–1932, 2004.
8. L. Xu and E. Oja. Randomized hough transform (rht): Basic mechanism, algorithm, and computational complexities. *CVGIP: Image Understanding*, vol. 57, pp. 131–154, 1995.
9. L. Xu, E. Oja and P.Klutanen, A new curve detection method: randomized Hough transform (RHT), *Pattern Recognition Letters*, vol. 11, 1990, pp. 331–338.
10. T.C. Chen and K.L. Chung. A new randomized algorithm for detecting lines. *Real-Time Imaging* , vol. 7, no. 6, pp. 473–481, 2001.
11. J. Canny. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, No. 6, 1986.
    E. Davies
12. Intel Corporation. Open source computer vision library.
    http://www.intel.com/technology/computing/opencv/index.htm

Guided Importance Sampling Based Particle Filtering for Visual Tracking

Kazuhiko Kawamoto

Kyushu Institute of Technology,
1-1 Sensui-cho, Tobata-ku, Kitakyushu 804-8550, Japan
kawa@mns.kyutech.ac.jp

Abstract. Linear estimation based sequential importance sampling methods for particle filters are proposed that can be used to model the rapid change of object motion in a video sequence. First a linear least–squares estimation is used to build a proposal function from observations, and then it is extended to a robust linear estimation. These sampling methods give a framework for tracking objects whose motion cannot be well modeled by a prior model. Finally a switching algorithm between the proposed method and the prior model based sampling method is proposed to achieve a filtering of both smooth and rapid evolution of the state. The ability of the proposed method is illustrated on a real video sequence involving a rapidly moving object.

1 Introduction

Visual tracking is the problem of estimating the motion of an object in an image sequence. State space approaches to visual tracking are attractive because they do not suffer from local minimum problems and incorporate prior knowledge available into the system [1]. The Kalman filter has been widely used for visual tracking with linear and Gaussian state space models [2,3]. Over the last decade, nonlinear and non–Gaussian state space methods with Monte Carlo approximation [4,5,6,7], called *particle filter*, has attracted much attention in visual tracking (e.g. [8,9,10]), because this approach allows to flexibly describe the state of targets with a wide range of models and gives an approximation of optimal solution from a Bayesian statistics point of view.

The basic idea of particle filters is to approximately represent the filtering distribution with a finite number of samples, referred to as *particle*, and to propagate the set of the particles according to an appropriate probability distribution over time. The most common probability distribution used for propagation in visual tracking is the prior model $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, which describes the system dynamics, because it is simply implemented and this may be because the pioneer works [4,5,8] use $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. However this often gives a poor estimate if the state transition which cannot be well modeled by the prior model occurs, which situation often happens due to rapid motion change of an object of interest.

We propose two proposal distributions, referred to as importance function, to track a rapidly moving object. The proposal distributions *guide* particles

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 158–167, 2006.

to give a good approximate of the filtering distribution. To do this, a linear estimate of motion parameters, which describes the state of the object, is first calculated from optical flows and then particles are generated from a distribution conditional on the linear estimate. Our motivation relies on the assumption that the linear estimate gives a rough estimate of the mode of the filtering distribution. First we present a proposal distribution based on a linear least–squares method and then extend it to a robust linear estimation method. Finally we present a switching algorithm between the proposed distribution and the prior distribution to achieve both smooth and rapid motion estimation by detecting rapid change of motion.

2 Particle Filtering for Visual Tracking

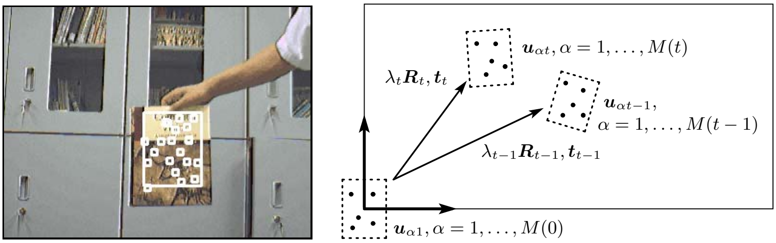
This paper focuses on the state space approach for visual tracking, which estimates recursively in time the filtering distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ of the state $\boldsymbol{x}_t \in \boldsymbol{R}^{n_x}$, given the sequence of observations $\boldsymbol{y}_{1:t} \equiv \{\boldsymbol{y}_k | k = 1, 2, \ldots, t\}, \boldsymbol{y}_k \in \boldsymbol{R}^{n_y}$. The states $\boldsymbol{x}_k, k = 1, 2, \ldots, t$ are assumed to be Markovian given an initial distribution $p(\boldsymbol{x}_0)$ and a transition distribution $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. The observations $\boldsymbol{y}_k, k = 1, 2, \ldots, t$, are conditionally independent of distribution $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ given the state. A recursive estimation for $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ consists of two steps: prediction $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}$ and filtering $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$

Particle filters give a numerical solution to this recursive estimation with Monte Carlo methods. The basic idea of particle filtering is that one approximately represents the filtering distribution in the pointwise form $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^{(i)}), \sum_{i=1}^{N} w_t^{(i)} = 1$, where $\delta(\cdot)$ denotes the delta-Dirac function. $\boldsymbol{x}_k^{(i)}$ is independent and identically distributed random sample, called *particle*, according to $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, and $w_t^{(i)}$ is the normalized weight associated with $\boldsymbol{x}_k^{(i)}$.

It is usually impossible to efficiently sample points from $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$. An alternative solution is the use of sequential importance sampling [7]. Instead of sampling from $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ directly, a so-called proposal distribution (or referred to as importance function), denoted by $\pi(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, is used to mimic random samples from $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$. The proposal distribution $\pi(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ leads to the update rule of the weight

$$w_t^{(i)} = \frac{w_t^{*(i)}}{\sum_{i=1}^{N} w_t^{*(i)}}, \quad w_t^{*(i)} = w_{t-1}^{(i)} \frac{p(\boldsymbol{y}_t|\tilde{\boldsymbol{x}}_t^{(i)})p(\tilde{\boldsymbol{x}}_t^{(i)}|\tilde{\boldsymbol{x}}_{t-1}^{(i)})}{\pi(\tilde{\boldsymbol{x}}_t^{(i)}|\tilde{\boldsymbol{x}}_{t-1}^{(i)}, \boldsymbol{y}_{1:t})}. \tag{1}$$

There are many choices of proposal distributions, dependent on applications of interest. The widely used proposal distribution is the prior distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, which describes the system dynamics [5,4,8]. In this case eq. (1) reduces to $w_t^{*(i)} = w_{t-1}^{(i)} p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)})$ and the algorithm is then easily implemented. As system models for visual tracking, low–dimensional autoregressive models such as $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{v}_t$ and $\boldsymbol{x}_{t+1} = 2\boldsymbol{x}_t - \boldsymbol{x}_{t-1} + \boldsymbol{v}_t$ are widely used [9], where $\boldsymbol{v}_t$ is a random noise,


$\boldsymbol{u}_{\alpha t}, \alpha = 1, \ldots, M(t)$
$\lambda_t \boldsymbol{R}_t, \boldsymbol{t}_t$
$\boldsymbol{u}_{\alpha t-1}$,
$\alpha = 1, \ldots, M(t-1)$
$\lambda_{t-1} \boldsymbol{R}_{t-1}, \boldsymbol{t}_{t-1}$
$\boldsymbol{u}_{\alpha 1}, \alpha = 1, \ldots, M(0)$

(a) (b)

Fig. 1. (a)Region of interest selected by hand and feature points detected in the region. (b)Motion model by similar transformation.

because one does not normally have specific knowledge on object motion. These system models however cannot deal with rapid motion change of an object, because they trend to smoothly estimate the state trajectory.

Hence an alternative proposal distribution is necessary. ICondensation [11] constructs a proposal distribution by detecting skin color regions in the image for hand tracking. Specific knowledge about object appearance is however not always available for a wide range of objects.

3 Motion and Observation Models

This section specifies the motion and observation models used in this paper. We represent an object of interest by its bounding box and assume some feature points in the box are tracked by an appropriate image processing through the image sequence, as shown in Fig. 1 (a). In experiments, we use the Lucas–Kanade–Tomasi tracker [13] with pyramidal implementation [14] for point tracking.

We model the motion of the object as the similar transformation. Since the similar transformation is parameterized by $(\theta, \lambda, t_1, t_2)$, the state vector $\boldsymbol{x}_t$ at time t is defined by $\boldsymbol{x}_t = (\theta(t), \lambda(t), t_1(t), t_2(t))^\top$. Assuming that the state evolves smoothly with time, we adopt a simple system model

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{v}_t, \quad \boldsymbol{v}_t \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}_v), \tag{2}$$

where $\boldsymbol{v}_t$ is a white Gaussian noise with mean $\boldsymbol{0}$ and covariance matrix $\boldsymbol{\Sigma}_v = \mathrm{diag}(\sigma_\theta^2, \sigma_\lambda^2, \sigma_{t_1}^2, \sigma_{t_2}^2)$. Of course this simple model can be replaced with other models. However, in visual tracking, the best model cannot be available because the sequence of the state is normally nonstationary and changeable. Hence we use this simple model.

We denote the αth feature point at time t by $\boldsymbol{y}_{\alpha t} = (x_\alpha(t), y_\alpha(t))^\top, \alpha = 1, \ldots, M(t)$, where $M(t)$ is the number of the feature points at time t. For notational simplicity, we denote the set of the observations by $\boldsymbol{y}_t \equiv \{\boldsymbol{y}_{1t}, \ldots, \boldsymbol{y}_{M(t)t}\}$. We define the observation model by a mixture distribution

$$p(\boldsymbol{y}_t | \boldsymbol{x}_t) \propto \frac{1}{M(t)} \sum_{\alpha=1}^{M(t)} \exp\left(-\frac{1}{2} (\boldsymbol{y}_{\alpha t} - \boldsymbol{u}_{\alpha t})^\top \boldsymbol{\Sigma}_w^{-1} (\boldsymbol{y}_{\alpha t} - \boldsymbol{u}_{\alpha t}) \right),$$

where $\boldsymbol{\Sigma}_w = \mathrm{diag}(\sigma_x^2, \sigma_y^2)$ and $\boldsymbol{u}_{\alpha t} = (u_\alpha(t), v_\alpha(t))^\top$ is obtained by the similar transformation $\boldsymbol{u}_{\alpha t} = \lambda(t)\boldsymbol{R}_t\boldsymbol{u}_{\alpha 1} + \boldsymbol{t}_t$. Note that $\boldsymbol{u}_{\alpha 1}$ is the position of the αth feature point detected in the first frame, and is moved in such a way that the center of the bounding box in the first frame is coincided with the origin of an image coordinate, as shown in Fig.1 (b).

4 Importance Sampling Guided by Linear Estimation

The prior distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ explores the state space without any knowledge of the observation $\boldsymbol{y}_t$, because $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is independent of $\boldsymbol{y}_t$. This characteristic is not appropriate for rapidly changing time–series data. An solution to this problem is the use of a proposal distribution conditional on observation. We propose two proposal distribution conditional on the observation $\boldsymbol{y}_t$, formally expressed by $\pi(\boldsymbol{x}_t|\boldsymbol{y}_t)$. First we present a linear least–squares estimation based distribution and then we extend it to a linear robust estimation.

4.1 Proposal Distribution Based on Least–Squares Estimation

We first build an overdetermined system of linear equations

$$\begin{pmatrix} u_1(1) & -v_1(1) & 1 & 0 \\ v_1(1) & u_1(1) & 0 & 1 \\ & \vdots & & \\ u_{M(1)}(1) & -v_{M(1)}(1) & 1 & 0 \\ v_{M(1)}(1) & u_{M(1)}(1) & 0 & 1 \end{pmatrix} \begin{pmatrix} s_{11}(t) \\ s_{12}(t) \\ t_1(t) \\ t_2(t) \end{pmatrix} = \begin{pmatrix} x_1(t) \\ y_1(t) \\ \vdots \\ x_{M(1)}(t) \\ y_{M(1)}(t) \end{pmatrix} \quad \text{or} \quad \boldsymbol{M}_t\boldsymbol{a}_t = \boldsymbol{b}_t, \quad (3)$$

where $s_{11} = \lambda\cos\theta$ and $s_{12} = -\lambda\sin\theta$. Calculating $\theta = \tan^{-1}(s_{11}/s_{12})$ and $s = \sqrt{s_{11}^2 + s_{12}^2}$, we obtain the linear estimate, denoted by $\hat{\boldsymbol{x}}_t = (\hat{\theta}(t), \hat{s}(t), \hat{t}_1(t), \hat{t}_2(t))^\top$, from the observations $\boldsymbol{y}_t$. Using $\hat{\boldsymbol{x}}_t$, we define the proposal distribution by

$$\pi(\boldsymbol{x}_t|\boldsymbol{y}_t) \equiv N(\hat{\boldsymbol{x}}_t, \boldsymbol{\Sigma}_{\hat{x}}), \quad (4)$$

where $\boldsymbol{\Sigma}_{\hat{x}} = \mathrm{diag}(\sigma_{\hat{\theta}}^2, \sigma_{\hat{\lambda}}^2, \sigma_{\hat{t}_1}^2, \sigma_{\hat{t}_2}^2)$. This proposal distribution generates particles in the vicinity of $\hat{\boldsymbol{x}}_t$, i.e., according to object motion, because $\hat{\boldsymbol{x}}_t$ is dependent on the observation. If $\hat{\boldsymbol{x}}_t$ is near the mode of the filtering distribution, the particles can well approximate the filtering distribution.

The computational cost depends on the cost of solving Eq. (3). If the number of the feature points $M(t)$ is constant over time, i.e., all of the points detected in the first frame are successfully tracked through the image sequence, the matrix $\boldsymbol{M}_t$ is also constant with respect to time. Therefore if we calculate the generalized inverse matrix of $\boldsymbol{M}_t$, denoted by $\boldsymbol{M}_t^-$, at time 1, it is only necessary to calculate $\boldsymbol{a}_t = \boldsymbol{M}_t^-\boldsymbol{b}_t$ at each time step. This is done more efficiently. If some of the points are invisible or lost, we need only to recalculate $\boldsymbol{M}_t^-$ at that time.

4.2 Proposal Distribution Based on Robust Estimation

Generally the least-squares methods are sensitive to outliers. Thus the previous approach gives a poor estimate if some of the feature points fail to be tracked. To deal with this problem, we introduce a RANSAC (Random Sample Consensus) [15] –like robust estimation method.

Our robust estimation method is as follow. Instead of using all of the feature points which may contain outliers to calculate the linear estimate, we randomly select the minimum number of the feature points that uniquely determines similar transformation, i.e., in this case two points, and calculate the parameters of similar transformation from these two points. Then we iterate this estimation a number of times. With an appropriate number of iterations, we expect to obtain at least one estimate free from outliers. To ensure this with probability γ, the number of iterations is set to

$$K = \frac{\log(1-\gamma)}{\log(1-(1-\epsilon)^2)}, \tag{5}$$

where ϵ is the outlier proportion. Finally, we obtain K linear estimates $\hat{\boldsymbol{x}}_{1t}, \ldots, \hat{\boldsymbol{x}}_{Kt}$ at time t, at least one of which is free from outliers with probability γ.

From $\hat{\boldsymbol{x}}_{1t}, \ldots, \hat{\boldsymbol{x}}_{Kt}$, we define the proposal distribution by a mixture distribution

$$\pi(\boldsymbol{x}_t|\boldsymbol{y}_t) \equiv \frac{1}{K}\sum_{\beta=1}^{K} N(\hat{\boldsymbol{x}}_{\beta t}, \boldsymbol{\Sigma}_{\hat{x}}). \tag{6}$$

Unlike RANSAC, we do not search the best fitting estimate, because this is likely to lead to a loss of the diversity of particles. This mixture distribution generates some particles in the vicinity of the linear estimate which are calculated from outlier-free observations. On the other hand, it generates some particles in the tail of the filtering distribution because some of $\hat{\boldsymbol{x}}_{1t}, \ldots, \hat{\boldsymbol{x}}_{Kt}$ are wrong estimates due to outliers. However such particles can be removed by resampling because the weights are low. The construction of the proposal distribution requires solving K systems of linear equations in four unknowns but does not need numerical search.

4.3 Combination of Smooth and Rapid Motion Estimation

The proposed proposal distribution $\pi(\boldsymbol{x}_t|\boldsymbol{y}_t)$ is superior to tracking a rapidly moving object than the prior distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, as discussed above. However, when the object is slowly moving, i.e., the state transition is well modeled by $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, the prior distribution can give a good approximation of the filtering distribution. Contrary, an estimate from $\pi(\boldsymbol{x}_t|\boldsymbol{y}_t)$ trends to overfit the state sequence corrupted by noise, because samples from $\pi(\boldsymbol{x}_t|\boldsymbol{y}_t)$ are generated only based on the observation. To sum up, a switching algorithm between $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $\pi(\boldsymbol{x}_t|\boldsymbol{y}_t)$ is preferable; $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is used when the object is slowly moving and $\pi(\boldsymbol{x}_t|\boldsymbol{y}_t)$ is used when the object is rapidly moving.

To detect rapid motion change of the object, we use an estimate of the effective sample size [6]

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N}(w_t^{(i)})^2} \leq N. \tag{7}$$

This is normally used to detect degeneracy phenomena in which almost all of the weights are close to zero with time. A degeneracy phenomenon can be detected by thresholding $\hat{N}_{eff} < N_T$ because $\hat{N}_{eff}$ becomes small if such a phenomenon happens. Here we use it as an indicator of detecting rapid motion change.

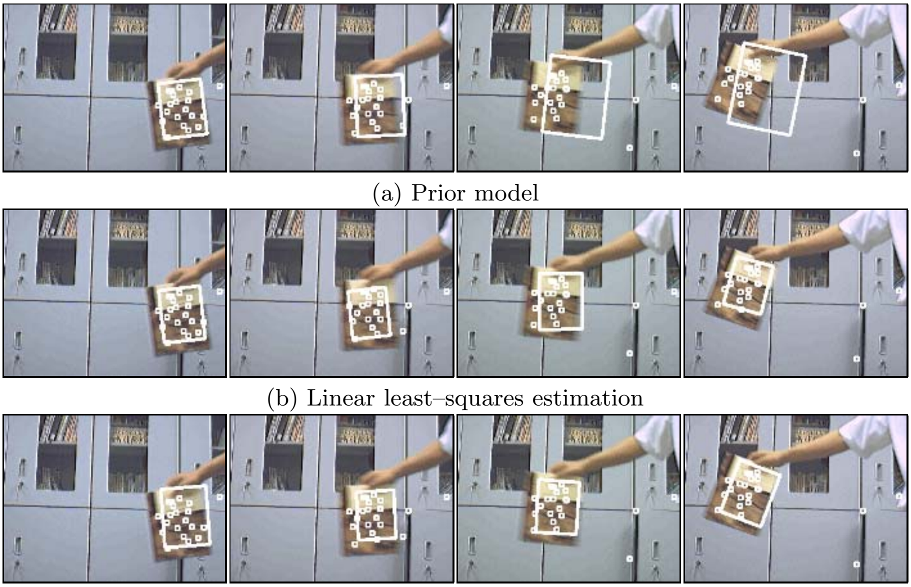
The procedure is as follows. First one tries to make a predictive distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ by choosing the prior distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ as a proposal distribution. If the prediction is correct, a degeneracy phenomenon does not happen, i.e., the value of $\hat{N}_{eff}$ is not small. Otherwise it is recognized that the prediction fails and then one tries to use the particles from $\pi(\boldsymbol{x}_t|\boldsymbol{y}_t)$ to estimate the filtering distribution. To sum up, we show an switching algorithm between $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $\pi(\boldsymbol{x}_t|\boldsymbol{y}_t)$ as follow.

```
1.  Initialization (t = 0):
    – For $i = 1, 2, \ldots, N$ draw $\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$, and set $t \leftarrow 1$.
2.  Importance sampling ($t \geq 1$):
2.1 Prior model based sampling
    – For $i = 1, 2, \ldots, N$ draw $\tilde{\boldsymbol{x}}_t^{(i)} \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i)})$ and calculate the weight by
      $w_t^{*(i)} = w_{t-1}^{(i)} p(\boldsymbol{y}_t|\tilde{\boldsymbol{x}}_t^{(i)})$.
    – For $i = 1, 2, \ldots, N$ normalize the weight $w_t^{(i)} = w_t^{*(i)} / \sum_{i=1}^{N_x} w_t^{*(i)}$.
2.2 Rapid change detection
    – Calculate $\hat{N}_{eff}$ by eq. (7).
    – If $\hat{N}_{eff} > N_T$, go to 3.
2.3 Linear estimation based sampling
    – For $i = 1, 2, \ldots, N$ draw $\tilde{\boldsymbol{x}}_t^{(i)} \sim \pi(\boldsymbol{x}_t|\boldsymbol{y}_t)$ and calculate the weight $w_t^{(i)}$ by
      eq. (1).
3.  Resampling:
    – For $i = 1, 2, \ldots, N$ resample with replacement the particles $\boldsymbol{x}_t^{(i)}$ from $\{\tilde{\boldsymbol{x}}_t^{(i)}|i =
      1, \ldots, N\}$ according to $w_t^{(i)}$ and set $w_t^{(i)} = 1/N$
    – $t \leftarrow t + 1$, go to 2..
```

---

5 Experimental Comparison with Real Video Sequence

Using a real video sequence involving a rapidly moving object, we compare the three proposal distributions: (a) the prior distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, (b) the linear least–squares estimation based distribution in Eq. (4), and (c) the robust linear estimation based distribution in Eq. (6). Finally we show the result of the switching algorithm, described in Sec. 4.3.



(a) Prior model

(b) Linear least–squares estimation

(c) Robust linear estimation

Fig. 2. Mean estimate of the state with the three proposal distributions

5.1 Experimental Setting

The video sequence consists of 450 frames (15s) at 320 × 240 pixels resolution, in which a book is moved quickly by hand. Figures 2 show some examples of the video sequence. In the first frame, feature points are detected by the Harris detector [12] within a manually selected region, shown in Fig. 1(a). Then the feature points are tracked through the video sequence by the Lucas–Kanade–Tomasi tracker [13] with pyramidal implementation [14]. This implementation can deal with large displacements of feature points using pyramidal search.

The number of particles is set to 2000. The variances of the system and the observation noises are set to $\sigma_\theta = 3.0\,(\text{degree}), \sigma_\lambda = 0.1, \sigma_{t_x} = \sigma_{t_y} = 3.0\,(\text{pixel})$ and $\sigma_x = \sigma_y = 3.0\,(\text{pixel})$, respectively. For the robust estimation method, the probability γ and the outlier proportion ϵ are set to $\gamma = 0.95$ and $\epsilon = 0.3$, i.e., the iteration number K is 5 from Eq. (5). In this setting, even if 30% of the feature points are outliers, we obtain at least one robust estimate with probability 0.95. For the switching algorithm, the threshold $\hat{N}_T$ is set to $\hat{N}_T = N/10 = 200$. The algorithms are implemented on a computer with Pentium 4 (3.4GHz) and 2 GB main memory.

5.2 Comparison of the Three Proposal Distributions

Figures 2 (a), (b), and (c) show the mean estimates (the white rectangles) of the particles at frames 70, 75, 80, 85 in a left-to-right fashion. These results are

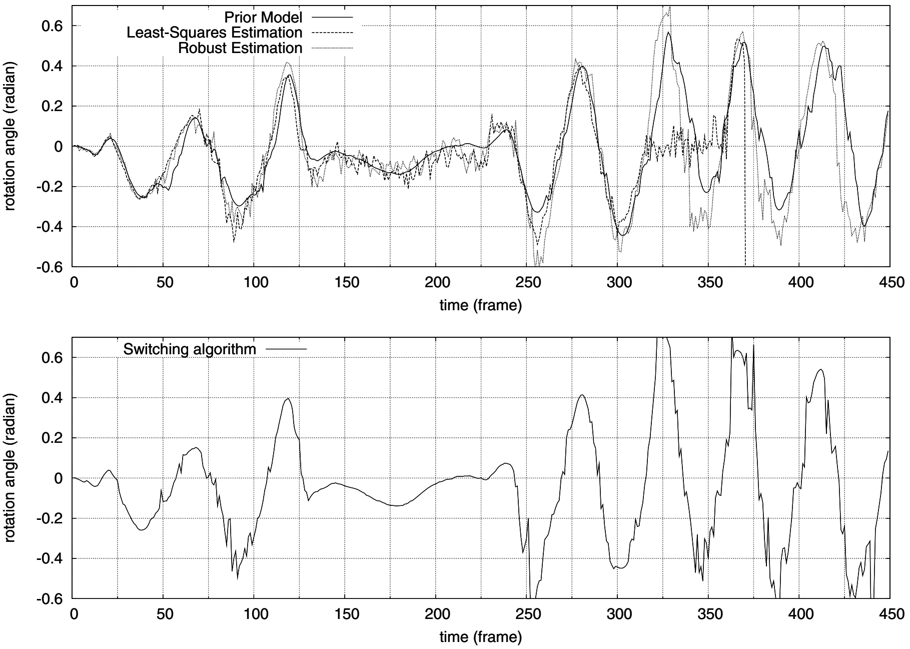


Fig. 3. Trajectory of the mean estimate for rotational angle with (top) the three proposal distributions and (bottom) by the switching algorithm

selected to show typical difference between the prior model and the proposed distributions, because around these frames the book is rapidly moving from right to left. Figures 2 (a) shows that the prior model based sampling method does not capture the rapid object motion, whereas the proposed methods do successfully. Figures 3 and 4 show the trajectories of the mean estimates for the rotational angle and the scale, respectively, of the state (owing to limited space, we do not show the results of the translational parameters here).

From Fig. 3 (top) and Fig. 4 (top), we confirm the prior model, labeled by "Prior Model", smoothly estimates the state trajectory around 75 frame, which correspond to Fig. 2, i.e., the prior model fails to correctly model the rapid motion of the object. We also find similar results around 140, 260, 330, 350, and 380 frames when the book is swing quickly. On the contrary, the proposed methods, labeled by "Least-Squares Estimation" and "Robust Estimation", trends to overfit the state trajectory corrupted by noise in Fig. 3 (top) and Fig. 4 (top). In addition, after around 390 frame, the linear least-squares estimation based sampling method fails to track the object because of outliers. These results with the three proposal distributions show their distinctive characteristics. The average execution times per frame are (a) the prior distribution: 5.17ms, (b) the linear least–squares estimation: 5.24ms, and (c) the robust linear estimation: 5.39ms.

Finally we show the results by the switching algorithm in Fig. 3 (bottom) and Fig. 4 (bottom). In the algorithm, we use the robust estimation based distribution

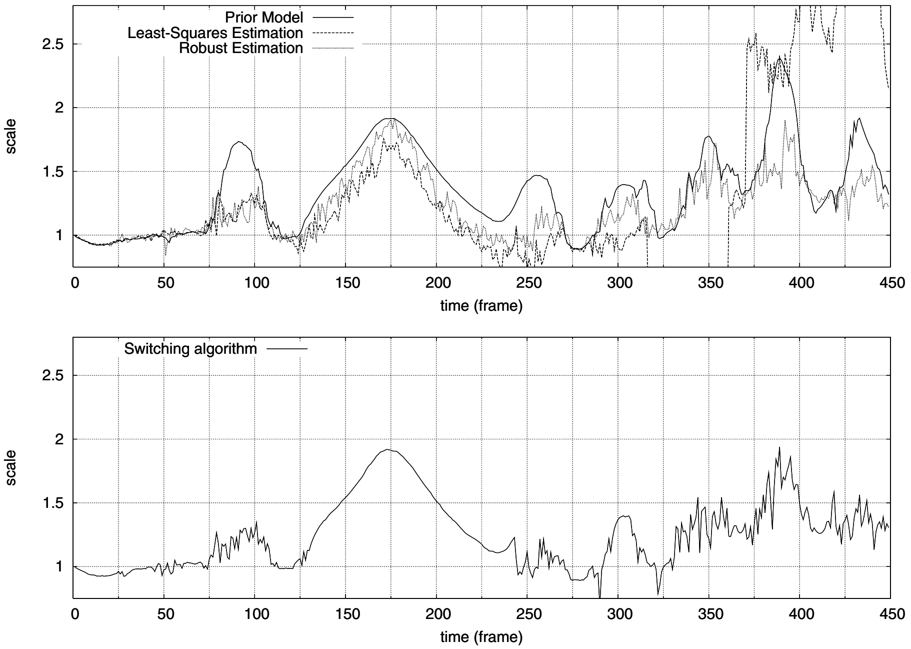


Fig. 4. Trajectory of the mean estimate for scale with (top) the three proposal distributions and (bottom) by the switching algorithm

in Eq. (6) as the proposal distribution. These results shows that the switching algorithm achieves smooth and rapid motion estimation, compared to those of the other methods in Fig. 3 (top) and Fig. 4 (top). The average execution times per frame is 6.26ms.

6 Conclusions

We have proposed the two proposal distribution for tracking rapidly moving objects in image sequences and the switching algorithm for smooth and rapid motion estimation. The optimal proposal distribution for sequential importance sampling is $\pi(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \equiv p(\boldsymbol{x}_t|\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t)$ which minimizes the variance of the weight w_t [7]. The analytical evaluation is impossible except a few special models. Our motivation is to provide an alternative solution. We design the proposal distributions conditional on the observation, whereas the prior distribution does not depend on the observation. This provides an advantage of tracking a rapidly moving object. Both the linear least-squares and the robust linear estimations are carried out by solving the system of linear equations. It is not difficult to implement the procedure on a computer because there are many program libraries and packages for linear algebra available. This fact yields practical benefit. We also have proposed the switching algorithm between the prior model and the proposed proposal distributions to combine the strengths of them.

Our approach can be extended to 3D motion estimation in a straightforward manner. This is a future work. In addition it is easy to use other image features such as contour and color together with feature points. This combination can improve the robustness of the tracking algorithm.

Acknowledgements. This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant-in-Aid (No.16700169).

References

1. D. Koller, K. Daniilidis, and H.-H. Nagel, Model-based object tracking in monocular image sequences of road traffic scenes, Int. J. Computer Vision, vol.10, pp.257–281, 1993.
2. L. Matthies, R. Szelinski, and T. Kanade, Kalman Filter-based Algorithms for Estimating Depth from Image Sequences, Int. J. Computer Vision, vol.3 (3), pp. 209–238, 1989.
3. D. B. Gennery, Visual tracking of known three-dimensional objects, Int. J. Computer Vision, vol.7 (3), pp. 243–270, 1992.
4. N. J. Gordon, D. J. Salmond, and A. F. M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proc.–F, vol.140 (2), pp. 107–113, 1993.
5. G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, J. Comput. Graph. Stat., vol.5, no.1, pp. 1–25, 1996.
6. J. S. Liu and R. Chen, Sequential Monte Carlo methods for dynamical systems, J. of the American Statistical Association, vol. 93, no. 443, pp. 1032–1044, 1998.
7. A. Doucet, S. Godsill and C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and Computing, vol. 10, pp. 197–208, 2000.
8. M. Isard and A. Blake, Condensation – Conditional density propagation for visual tracking, Int. J. Computer Vision, vol.29 (1), pp.5–28, 1998.
9. N. Ichimura and N. Ikoma, Filtering and Smoothing for Motion Trajectory of Feature Point Using Non-Gaussian State Space Model, IEICE Trans. Inf. & Syst., vol.E84-D, no.6, pp.755–759, 2001.
10. K. Nummiaro, E. Koller-Meier, and L. V. Gool, An adaptive color-based particle filter, Image and Vision Computing, vol.21, pp.99–110, 2003.
11. M. Isard and A. Blake, "ICondensation: Unifying low-level and high-level tracking in a stochastic framework", Proc. 5th ECCV, vol.1, pp.893–908, 1998.
12. C. Harris and M. Stephens, A combined corner and edge detector, Proc. 4th Alvey Vision Conf., pp.147–151, Aug. 1988.
13. J. Shi and C. Tomasi, Good Features to Track, Proc. CVPR, pp.593–600, 1994.
14. J. Y. Bouguet, Pyramidal Implementation of the Lucas Kanade Feature Tracker, Intel Corporation, Microprocessor Research Labs, 2000,
15. M. A. Fischer and R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Comm. ACM, vol. 24 (6), pp. 381–395, 1981.

Intelligent Spot Detection for 2-DE Gel Image*

Yi-Sheng Liu[1], Shu-Yuan Chen[1], Ya-Ting Chao[1], Ru-Sheng Liu[1],
Yuan-Ching Tsai[2], and Jaw-Shu Hsieh[2]

[1] Department of Computer Science and Engineering, Yuan Ze University,
Chung Li, Taiwan
s889404@mail.yzu.edu.tw, {cschen, ytchao, csrobinl}@saturn.yzu.edu.tw
[2] Department of Agronomy, National Taiwan University, Taipei, Taiwan
botsai@gate.sinica.edu.tw, jawar@ntu.edu.tw

Abstract. In this study, a novel method for spot detection is proposed with the addition of confidence evaluation for each detected spot. The confidence of a spot will give useful hints for subsequent processing such as landmark selection, spot quantification, gel image registration, etc. The proposed method takes slices of a gel image in the gray level direction and build them into a slice tree, which in turn is used to perform spot detection and confidence evaluation. Moreover, the proposed method is fast. Building slice tree for a gel image of 1262×720 take about 3.2 sec. Spot detection take about 66 ms after the slice tree was built. Experimental results show that confidence values are close to subjective judgement.

Keywords: 2-DE gel, Protein, Spot detection, Slice tree.

1 Introduction

Proteomics is the study of proteme, especially how the proteins are functioning in and around cells. Protein separation is one of the most important stages in the proteomics study. Among all separation techniques, two-dimensional electrophoresis (2-DE) [1,2,3,4] is the best method to separate complex protein mixtures according to their charge and size. Spots in the gel are proteins migrated to specific locations. According to the differential expression of protein mixtures from control and experimental samples, the spots in gel may disappear, appear or chang in size and intensity. By analysis of spot appearance in gel, differential protein expression between various samples are obtained.

Due to the volume data and technical noise originating from the image acquisition process, manual analysis of gel image is difficult without the help of computer software. Analysis of gel image by image processing software requires an image pipeline which may contain image correction, spot detection, spot

* This work was partially supported by the National Science Council of Taiwan, R.O.C., under Grants NSC-95-2745-E-155-008-URD.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 168–177, 2006.

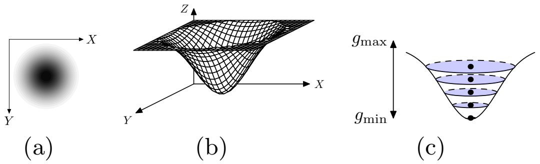


Fig. 1. Slices of spot. (a) A spot in gel image. (b) A spot in three dimension view. (c) Slices of spot and corresponding central points.

quantification, spot registration, data presentation and interpretation. Quality of spot detection is one of the most important factors that will influence the performance of the image pipeline. A comprehensive review of those computation techniques can be referred to [3].

In this study, a novel method for spot detection is proposed with the addition of confidence calculation for each detected spot. The confidence of a spot will give useful hints for subsequent processing such as landmark selection, spot matching, gel image registration, etc. The proposed method takes slices of a gel image in the gray level direction and build them into a slice tree, which in turn is used to perform spot detection and confidence calculation.

This paper is organized as follows. The idea to detect spots in gel images by slice tree is presented in Section 2. The detailed description about our algorithm is presented in Section 3. Finally, experimental results and conclusions are given in Sections 4 and 5, respectively.

2 Approach

The key concept of the proposed approach is introduced in this section. Unlike other spot detection method, our method slices the gel image, builds them into a slice tree, and then detects spots on the basis of slice tree.

Let the intensity be the third-dimension (Z axis), the intensity of a sliver-stained spot is approximately Gaussian distributed with the lowest intensity at center as shown in Fig. 1(b). A series of slices of the spot can then be obtained in the intensity direction as shown in Fig. 1(c). Each slice has its own features such as size, shape, central point, boundary smoothness and so on. If we project the central points of slices onto the X-Y plan. The projected points belong to the same spot will fall in a neighborhood. The distribution and the number of projected points depend on the shape and appearance of the spots in a gel image which can be used for spot detection.

In fact, spots may be distorted [1,5], overlapping [6] and suffered with noise. These factors make spot detection more difficult and un-reliable. The relationship between the slices of the spots can be incorporated into the slice tree so as to resolve these problems and then obtain a robust spot detector.

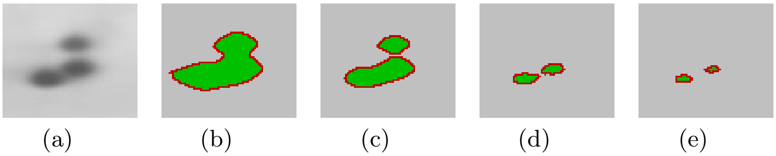


Fig. 2. Border tracing to detect regions in binary images. (a) Sample gel image with gray levels from 90 to 218. (b)-(e) are detected results of B_{187}, B_{162}, B_{108} and B_{96} respectively. A border is a contour labeled by red color and all the pixels with green color inside the border compose the corresponding region.

3 Methods

3.1 Gel Image Slicing

For a 2D-gel image I, the binarized image B_g related to gray level g is defined by

$$B_g(x,y) = \begin{cases} 1 \text{ if } I(x,y) \le g, \\ 0 \text{ otherwise}, \end{cases} \tag{1}$$

where $I(x,y)$ is the intensity of pixel at coordinates (x,y) and g is one of the gray levels between the maximum and minimum gray levels of I, denoted by $g_{\max}$ and $g_{\min}$, respectively.

Definition 1. *Regions. Let r be a subset of pixels in a binary image. We call r a region in a binary image if r is a connected set.*

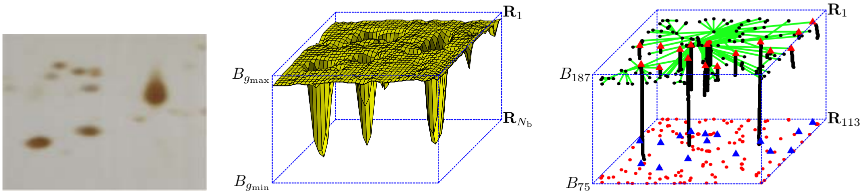
Regions can be detected in the following way. Region borders in a binary image are first detected by border tracing. The set of pixels enclosed by a border is then denoted as the corresponding region. Some results of border tracing and the detected regions are shown in Fig. 2. From this figure, we can also find that a candidate spot with minimum gray level $g_{s_{\min}}$ will appear as sequence of regions in binary images B_{g_s} for $g_{\max} \ge g_s \ge g_{s_{\min}}$. Intuitively, the sequence of binary images from $B_{g_{\max}}$ to $B_{g_{\min}}$ can be regarded as computerized tomography (CT) images of all the spots in the gray level direction, i.e. Z axis.

Definition 2. *Region set. All regions in a binary image is called a region set of the binary image.*

For the gel image I, there are $N_{\mathrm{b}} = g_{\max} - g_{\min} + 1$ binary images. We sort binary images B_g in the descending order of g and let $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_{N_{\mathrm{b}}}$ be the region sets related to the binary images $B_{g_{\max}}, B_{g_{\max}-1}, \ldots, B_{g_{\min}}$, respectively. i.e.

$$\mathbf{R}_s = \{r_{s,i} | i = 1, 2, \ldots, n_s\}, s = 1, 2, \ldots, N_{\mathrm{b}} \tag{2}$$

where $r_{s,i}$ are the regions in binary image $B_{g_{\max}+1-s}$ and n_s is the number of regions in the binary image $B_{g_{\max}+1-s}$. Note that the value of s can be regarded as the layer index of the region sets. A sample gel image to illustrate the relationship between B_g and $\mathbf{R}_s$ is shown in Fig. 3(b).



(a) (b) (c)

Fig. 3. Slice tree of gel image. (a) Sample gel image. (b) 3D view of (a), relationship between B_g and $\mathbf{R}_s$ are shown beside the cube. (c) Corresponding slice tree of (a). Central points of regions are shown by black circular points, parent and children region are connected by green links, detected spots are shown by triangular marks. Projections of region centers and spots are also shown in $B_{g_{min}}$.

3.2 Properties of Regions

Some properties of regions are described in this section.

Definition 3. *Binary image projection. For a binary image B_g, the projection of B_g, $\Psi(B_g)$, is defined as a set of coordinates whose corresponding pixel values are 1. i.e.*

$$\Psi(B_g) = \{(x, y)|B_g(x, y) = 1\} \quad . \tag{3}$$

Since a region is a subset of a binary image, the operation Ψ can also be applied to a region. i.e.

$$\Psi(r_{s,i}) = \{(x, y)|r_{s,i}(x, y) = 1\} \quad . \tag{4}$$

Definition 4. *Ancestor region and descendant region. For two regions $r_{s_1,i}$ and $r_{s_2,j}$, with $s_1 < s_2$, if $\Psi(r_{s_1,i}) \supseteq \Psi(r_{s_2,j})$, then we say $r_{s_1,i}$ is ancestor region of $r_{s_2,j}$ and $r_{s_2,j}$ is descendant region of $r_{s_1,i}$, and denoted by*

$$r_{s_1,i} \Supset r_{s_2,j} \quad . \tag{5}$$

Definition 5. *Child region and parent region. For two regions $r_{s_1,i}$ and $r_{s_2,j}$, if $s_1 = s_2 - 1$ and $r_{s_1,i} \Supset r_{s_2,j}$, then we say $r_{s_2,j}$ is a child region of $r_{s_1,i}$ and $r_{s_1,i}$ is the parent region of $r_{s_2,j}$.*

Property 1. All regions in a binary image are mutual exclusive. i.e.

$$\Psi(r_{s,i}) \cap \Psi(r_{s,j}) = \emptyset \text{ if } i \neq j \quad . \tag{6}$$

Property 2. Every region in $\mathbf{R}_2, \mathbf{R}_3, \ldots, \mathbf{R}_{N_\text{b}}$ has exactly one parent region. i.e. For $s = 2, 3, \ldots, N_\text{b}$

$$\forall r_{s,i} \in \mathbf{R}_s, \exists! r_{s-1,k} \in \mathbf{R}_{s-1}, \text{s.t. } \Psi(r_{s,i}) \subseteq \Psi(r_{s-1,k}) \quad . \tag{7}$$

Property 3. For two regions $r_{s,i}$ and $r_{s-1,k}$, $r_{s,i}$ is a child region of $r_{s-1,k}$ if and only if their projection are overlapping. i.e.

$$\Psi(r_{s,i}) \subseteq \Psi(r_{s-1,k}) \iff \Psi(r_{s,i}) \cap \Psi(r_{s-1,k}) \neq \emptyset \quad . \tag{8}$$

Property 3 will simplify the procedure of finding child regions. For a region $r_{s,i} \in \mathbf{R}_s$, the child regions of $r_{s,i}$ can be found in the area $\Psi(r_{s,i})$ of binary image $B_{g_{\max}-s}$. Let the set of all child regions of $r_{s,i}$ be denoted by $\mathbf{R}_{s,i}$, then it is obvious that $\mathbf{R}_{s+1} = \bigcup_{i=1}^{n_s} \mathbf{R}_{s,i}$. By Property 3, $\mathbf{R}_{s,i}$ can be constructed as

$$\mathbf{R}_{s,i} = \{r_{s+1,j} | \Psi(r_{s+1,j}) \cap \Psi(r_{s,i}) \neq \emptyset\}, j = 1, 2, \ldots, n_{s+1} \ . \tag{9}$$

3.3 Slice Tree

Regions in binary images are the basic units for spot detection and confidence calculation in our method. To increase the robustness of spot detection, the relationship between regions in successive binary images related to the same spot are organized in a slice tree.

Definition 6. *Slice tree. A slice tree for gel image I can be defined as $T = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is a set of nodes and $\mathbf{E}$ is a set of links between the nodes.*

Each node in the slice tree corresponds to a region. Hence, the node related to the region $r_{s,i}$ is denoted as $\mathcal{V}(r_{s,i})$. According to the layer structure of region sets in (2), $\mathbf{V}$ can be further divided into N_b exclusive subsets, that is

$$\mathbf{V} = \bigcup_{s=1}^{N_\mathrm{b}} \mathbf{V}_s \ . \tag{10}$$

Nodes in $\mathbf{V}_s$ correspond to regions in $\mathbf{R}_s$, i.e.,

$$\mathbf{V}_s = \{\mathcal{V}(r_{s,i}) | i = 1, 2, \ldots, n_s\} \ . \tag{11}$$

Note that nodes in $\mathbf{V}_s$ have depth $s - 1$ in the slice tree.

Each link in $\mathbf{E}$ is an ordered pair of nodes $(\mathcal{V}(r_{s-1,k}), \mathcal{V}(r_{s,i}))$, where $r_{s-1,k}$ is a parent region and $r_{s,i}$ is a child region, i.e.,

$$\mathbf{E} = \{(\mathcal{V}(r_{s-1,k}), \mathcal{V}(r_{s,i})) | r_{s-1,k} \Supset r_{s,i}\}, s = 2, \ldots, N_\mathrm{b} \ . \tag{12}$$

The slice tree for the sample gel image in Fig. 3(a) is shown in Fig. 3(c).

3.4 Slice Tree Construction

A slice tree for gel image I is constructed in the sequence of $B_{g_{\max}}, \ldots, B_{g_{\min}}$ accompanied by the establishment of relations between every two successive region sets $\mathbf{R}_s$ and $\mathbf{R}_{s+1}$ for $s = 1, 2, \ldots, N_\mathrm{b} - 1$. More specifically, the slice tree is built by recursively performing procedure **ProcessChildSlice**$(r_{s,i})$ with parameter as the region $r_{s,i}$ in $\mathbf{R}_s$. The pseudo code of the procedure is outlined as follow.

```
Procedure ProcessChildSlice($r_{s,i}$)
  1. Get child region set $\mathbf{R}_{s,i}$ of $r_{s,i}$ using (9).
  2. If $\mathbf{R}_{s,i} = \emptyset$ then return,
     else for all child regions $r_{s+1,j} \in \mathbf{R}_{s,i}$, do
     2.1 Create a tree node $\mathcal{V}(r_{s+1,j})$.
     2.2 Set parent-child link between $\mathcal{V}(r_{s,i})$ and $\mathcal{V}(r_{s+1,j})$.
     2.3 If $s < N_\mathrm{b} - 1$ then call ProcessChildSlice($r_{s+1,j}$).
```

To build slice tree for a gel image, we first create a root node $\mathcal{V}(r_{1,1})$, then call **ProcessChildSlice**$(r_{1,1})$. From (1), $r_{1,1} = B_{g_\mathrm{max}}$ covers the whole gel image and is the only region in $\mathbf{R}_1$.

3.5 Slice Tree Terminology

Let $N(\mathbf{R}_{s,i})$ denotes the number of child regions for $r_{s,i}$, then node $\mathcal{V}(r_{s,i})$ has $N(\mathbf{R}_{s,i})$ children in the slice tree. Nodes in slice tree can be divided into three categories according to the number of children:

1. Leaf nodes: $N(\mathbf{R}_{s,i}) = 0$.
2. Solitary nodes: $N(\mathbf{R}_{s,i}) = 1$.
3. Manifold nodes: $N(\mathbf{R}_{s,i}) > 1$.

Definition 7. *Sticks. If we remove all links between manifold nodes and their child nodes, the slice tree is divided into subgroups namely sticks.*

Note that all nodes in a stick have no more than one link. It is obvious that each spot in the gel image has a corresponding stick in the slice tree.

Definition 8. *Stick root, leaf stick, internal stick, sibling sticks, parent stick and child sticks. The node with minimum depth in a stick is called stick root. A stick is called a leaf stick if it contains leaf node, otherwise the stick is called an internal stick. Sticks whose stick roots have the same parent node in the original slice tree are called sibling sticks and also called child sticks of the stick where the parent node resides, which in turn is called parent stick.*

Definition 9. *Stick length. For a node $\mathcal{V}(r_{s,i})$, the stick length of $\mathcal{V}(r_{s,i})$ is defined as*

$$\mathcal{L}(r_{s,i}) = 1 - s + \arg\min_{d=s}^{N_\mathrm{b}} \left(N(\mathbf{R}_{d,i_d}) \neq 1 \right) \quad . \tag{13}$$

Note that for the case of $d = s$, the value of i_d is just i. For the other cases of d, i_d is an index such that $r_{d-1,i_{d-1}} \Supset r_{d,i_d}$. It is obvious that the stick length of a stick root is equal to the number of nodes in the stick and stick length of each leaf node is 1.

Definition 10. *Extended stick length. For a node $\mathcal{V}(r_{s,i})$, the extended stick length of $\mathcal{V}(r_{s,i})$ is defined as*

$$\mathcal{L}_\mathrm{e}(r_{s,i}) = 1 - s + \arg\max_{d=s}^{N_\mathrm{b}} \left(N(\mathbf{R}_{d,i_d}) = 0 \right) \quad . \tag{14}$$

The values of i_d are defined as those in (13). It is obvious that $\mathcal{L}(r) = \mathcal{L}_{\mathrm{e}}(r)$ if $\mathcal{V}(r)$ is a node in a leaf stick. Stick length of a stick is defined as the stick length of its stick root.

3.6 Spot Detection

Human recognize spots of gel image by size, shape and intensity variation of the spots. To utilize slice tree for spot detection, region size and stick length are used. The region size is expressed by the number of pixels in the region. If a region belong to a spot, it should have reasonable region size. Thus region size should be restricted to reduce image noises in spot detection. Stick length of each node in the slice tree corresponds to intensity gradient of the spot in gel image. A confident spot should have larger stick length but a faint spot has smaller stick length in the slice tree.

More specifically, spot detection using slice tree is done by performing two recursive procedures **FindSpotInTree**$(r_{s_{\mathrm{r}},i})$ and **ProcessStick**$(r_{s,j})$ where $\mathcal{V}(r_{s_{\mathrm{r}},i})$ is a stick root while $r_{s,j}$ could be $r_{s_{\mathrm{r}},i}$ or a descendant region of $r_{s_{\mathrm{r}},i}$. Three thresholds d, w_t and h_t are involved in the spot detection where d is the minimum stick length of a node to be recognized as a spot and w_t and h_t are the minimum width and height of a region which could be processed for spot detection. The sensitivity of spot detection can be controlled by adjusting the three threshold values. The pseudo code for spot detection is outlined as follow.

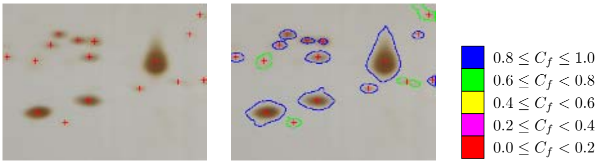
```
Procedure FindSpotInTree($r_{s_{\mathrm{r}},i}$)

  1. Call ProcessStick($r_{s_{\mathrm{r}},i}$).
  2. If no spots are found in all child sticks of $\mathcal{V}(r_{s_{\mathrm{r}},i})$ and $\mathcal{L}_{\mathrm{e}}(r_{s_{\mathrm{r}},i})$ is greater
     than or equal to $d$, then a spot is found at $\Psi(r_{s_{\mathrm{r}},i})$.

Procedure ProcessStick($r_{s,j}$)

  1. If width of $r_{s,j} \geq w_t$ and height of $r_{s,j} \geq h_t$
     then initiate $\mathcal{L}(r_{s,j}) = 1$,
     else initiate $\mathcal{L}(r_{s,j}) = 0$.
  2. If $N(\mathbf{R}_{s,j}) = 1$ then do
     2.1 For $r_{s+1,k} \in \mathbf{R}_{s,j}$, Call ProcessStick($r_{s+1,k}$).
     2.2 Set $\mathcal{L}(r_{s,j}) = \mathcal{L}(r_{s,j}) + \mathcal{L}(r_{s+1,k})$.
     2.3 Goto step 4.
  3. If $N(\mathbf{R}_{s,j}) > 1$ then
     for all $r_{s+1,k} \in \mathbf{R}_{s,j}$, call FindSpotInTree($r_{s+1,k}$).
  4. $\mathcal{L}_{\mathrm{e}}(r_{s,j}) = \mathcal{L}(r_{s,j}) + \max_{r \in \mathbf{R}_{s,j}}(\mathcal{L}_{\mathrm{e}}(r))$.
```

Parameters $r_{s_{\mathrm{r}},i}$ passed to **FindSpotInTree**$(r_{s_{\mathrm{r}},i})$ are regions corresponding to stick roots. **FindSpotInTree**$(r_{s_{\mathrm{r}},i})$ calls **ProcessStick**() to calculate stick length of $\mathcal{V}(r_{s_{\mathrm{r}},i})$ and check spot criteria for the node. If $\mathcal{V}(r_{s_{\mathrm{r}},i})$ belongs to a leaf stick and its stick length is greater than or equal to d, then a spot is found at $\Psi(r_{s_{\mathrm{r}},i})$. If no stick roots of sibling sticks satisfy the criteria, shorter sticks are pruned and the longest stick is merged with the parent stick, which in turn is



(a) (b) (c)

Fig. 4. Results of spot detection using slice tree. (a) The detected spots were marked with red crosses. (b) Confidence of detected spots are shown in various colors. (c) The mapping between confidence values and colors.

used for criteria testing. The pruning and merging procedure is repeated until a merged stick satisfies the criteria or the root node is reached.

ProcessStick($r_{s,j}$) checks the region size of $r_{s,j}$ and calculates stick length for node $\mathcal{V}(r_{s,j})$ by recursively calling itself with child region as parameter until a non-solitary node encountered. Those regions smaller than a specified size are eliminated during the calculation of stick length. When a manifold node is encountered, **FindSpotInTree**() is called to check spot criteria for child sticks originated from the manifold node. The results of spot detection for the gel image in Fig. 3(a) are shown in Fig. 4.

3.7 Confidence Evaluation for Spots

Since spots in the gel image have specific characteristics in the slice tree, their confidence can be computed by the features of the corresponding regions. More specifically, the confidence values of spots are computed on the basis of slice tree by the following equation.

$$C_f = \frac{\sqrt{(\alpha l)^2 + (\beta s)^2 + (\gamma c)^2}}{\alpha + \beta + \gamma} \tag{15}$$

where l, s and c are metrics for stick length, smoothness and compactness related to the spots, and α, β and γ are the respective weighting factors. If we identify spots by the regions where the spots have been detected, then the metrics are defined as follow.

$$l = \min(1.0, \frac{l_e}{\sqrt{n_\mathrm{p}}}) \tag{16}$$

$$s = \max(0.0, 1.0 - \delta \times \frac{n_\mathrm{r}}{n_\mathrm{b}}) \tag{17}$$

$$c = \min(1.0, \frac{2\sqrt{\pi n_\mathrm{p}}}{n_\mathrm{b}}) \tag{18}$$

where l_e is the extended stick length related to the spot, n_p is the number of region pixels, n_b is the number of border pixels of the region, n_r is the number of one-pixel-width knobs extended from the region and δ is a constant factor. The metrics are normalized to the range from 0 to 1. The larger are the metrics, the more confident spots are obtained.

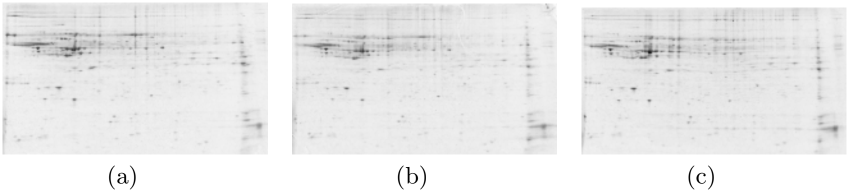


Fig. 5. Gel images [7] used for performance evaluation. (a) 031403-ctrl2.tiff (1262×724) (b) 031403-ctrl3.tiff (1262 × 720) (c) 031403-ctrl4.tiff (1262 × 700)

4 Experimental Results

The proposed method has been implemented on a notebook equipped with Intel Pentium III CPU 1.2GHz, 256MB RAM. The gel images [7] used for performance evaluation are shown in Fig. 5. The input gel images are first pre-processed by a 7 × 7 Gaussian filter. The average time took to build slice tree and detect spots for these images is 3.2 sec and 66 ms, respectively.

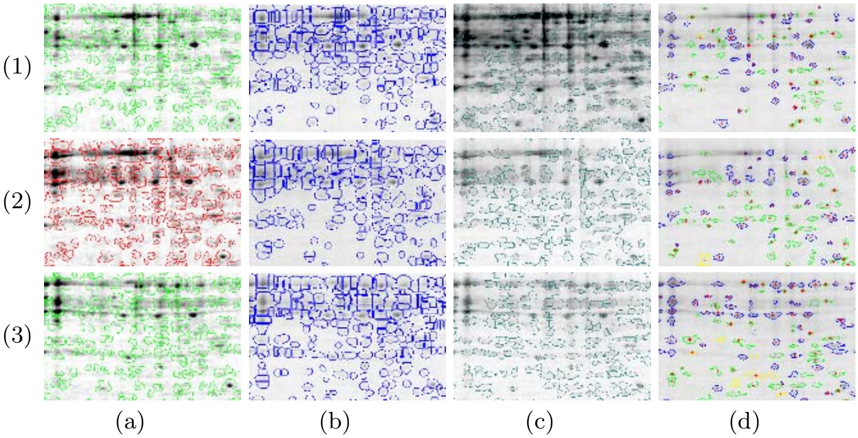


Fig. 6. Comparison of spot detection. Columns for (a) Delta2D, (b) Progenesis, (c) Proteomweaver and (d) Our method. (The mapping between confidence values and colors are as specified in Fig. 4(c)) Rows for (1) 031403-ctrl2.tiff, (2) 031403-ctrl3.tiff and (3) 031403-ctrl4.tiff.

To show the performance of spot detection using slice tree, the results of the proposed spot detection are compared to those of three commercial software [7]: Delta2D 3.2, Progenesis Discovery v.2005 and Proteomweaver 3.0.1.1. Most of the existing spot detection methods including the three methods adopted Watershed [8] algorithm for spot segmentation. After a gel image is segmented, spot models are employed to eliminate segments not being fitted by the model. Watershed is the most popular technique for spot segmentation, over segmentation is its well-known problem. Thus, the effectiveness of spot model are crucial to the results of spot detection based on watershed. Subblocks of 429 × 279 from the results of spot detection of the software packages are shown in the first three columns of Fig. 6.

Unlike other approaches, spot detection using slice tree does not relay on spot models. Instead, the stick length of each leaf stick corresponding to intensity difference between spots and background is used as the criterion of being confident spot. Results of our method are shown in the fourth column of Fig. 6. In our results, spot centers are marked with red crosses and the boundaries of spots are shown in different color according to their confidence values. It can be seen that the boundaries of spots in our method are more compact.

5 Conclusion

Slice tree is effective for representing a gel image in a systematic organization. Nodes in slice tree contain refined features about the spots and links between nodes contain corresponding characteristic expression of the gel image. Thus, gel image analysis can be done by analyzing the slice tree based on the systematic organization. In this paper, we have shown how to detect spots using slice tree. In addition slice tree with confidence evaluation can provide plentiful information for other applications such as spot quantification, gel image registration, etc. These will be the future research issues.

References

1. Aittokallio, T., Salmi, J., Nyman, T.A., Nevalainen, O.S.: Geometrical distortions in two-dimensional gels: applicable correction methids. Journal of Chromatography B **815** (2005) 25–37
2. Salmi, J., Aittokallio, T., Nyman, T.A., Nevalainen, O.S.: Correcting distortions in 2D-gels — a survey. Technical Report 653, Turku Centre for Computer Science (2004)
3. Dowsey, A.W., Dunn, M.J., Yang, G.Z.: The role of bioinformatics in two-dimensional gel electrophoresis. Proteomics **3** (2003) 1567–1596
4. Quadroni, M., James, P.: Proteomics and automation. Electrophoresis **20** (1999) 664–677
5. Kriegel, K., Seefeldt, I., Hoffmann, F., Schultz, C., Wenk, C., Regitz-Zagrosek, V., Oswald, H., Fleck, E.: An alternative approach to deal with geometric uncertainties in computer analysis of two-dimensional electrophoresis gels. Electrophoresis **21** (2000) 2637–2640
6. Pietrogrande, M.C., Marchetti, N., Dondi, F., Righetti, P.G.: Spot overlapping in two-dimensional polyacrylamide gel electrophoresis separations: A statistical study of complex protein maps. Electrophoresis **23** (2002) 283–291
7. Dunsmore, J.: Comparison of 2d gel spot detection algorithms. http://www.deltastat.org/2d-gel-algorithms-comparison.html (2005)
8. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Trans. Pattern Analysis and Machine Intelligence **13** (1991) 583–598

Gaze Estimation from Low Resolution Images

Yasuhiro Ono, Takahiro Okabe, and Yoichi Sato

Institute of Industrial Science, The University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
{onoy, takahiro, ysato}@iis.u-tokyo.ac.jp
http://www.hci.iis.u-tokyo.ac.jp/~onoy/

Abstract. The purpose of this study is to develop an appearance-based method for estimating gaze directions from low resolution images. The problem of estimating directions using low resolution images is that the position of an eye region cannot be determined accurately. In this work, we introduce two key ideas to cope with the problem: incorporating training images of eye regions with artificially added positioning errors, and separating the factor of gaze variation from that of positioning error based on N-mode SVD (Singular Value Decomposition). We show that estimation of gaze direction in this framework is formulated as a bilinear problem that is then solved by alternatively minimizing a bilinear cost function with respect to gaze direction and position of the eye region. In this paper, we describe the details of our proposed method and show experimental results that demonstrate the merits of our method.

Keywords: gaze estimation, low resolution, appearance-based method, positioning, N-mode SVD.

1 Introduction

Gaze direction is an important cue for understanding human activities since they are considered to be well correlated with our focus of attention. Thus robust and nonintrusive estimation of gaze direction, hereafter referred to as *gaze estimation*, can be used effectively for a wide variety of applications. For instance, gaze estimation techniques can be used for determining how often and which part of a billboard is being looked at in a public space such as a shopping mall.

One of the key challenges for gaze estimation for such applications is that it is not always possible to capture high resolution images due to limitation of camera placement. Therefore, it is important to have techniques for gaze estimation from *low resolution images*. For example, consider the case of estimating gaze directions by using images captured by a surveillance camera already installed in an environment. It is likely that the camera is far from a subject, and thus only low resolution images of the subject's face are available.

Previously proposed methods for gaze estimation, which are classified into two approaches: *model-based methods* and *appearance-based methods*, are not suitable for the purpose of gaze estimation from low resolution images for several reasons.

Model-based methods usually require high resolution images of human faces to estimate gaze direction accurately. This is because gaze directions are determined from the

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 178–188, 2006.

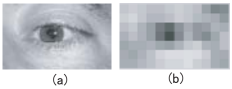


Fig. 1. Images of an eye with (a) high and (b) low resolutions. It is not trivial to accurately extract geometric features of the eye and feature points for positioning from the low resolution image.

eye's geometric features localized in images. Among model-based methods, the most commonly used techniques are the ones based on pupil corneal reflection [2,4,7,17,18]. A gaze direction is determined from the relative position of the pupil center and a glint reflected on the cornea of an eyeball. Other techniques use the position of an iris center or a pupil center obtained from edge detection or ellipse fitting for estimating gaze direction [5,6,14]. As we see in Figure 1, it is not trivial to extract the above features from low resolution images of an eye. In contrast, appearance-based methods can be used for estimating gaze directions from low resolution images because these methods use pixel values of eye regions for estimating gaze directions, and therefore it is not necessary to find the eye's geometric features in input images. Unlike model-based methods, appearance-based methods have had very few studies devoted to them. Some researchers have proposed gaze estimation methods using a neural network that is trained with eye images of known gaze directions [1,10,15]. Recently, Tan *et al.* developed a method based on nearest neighbor search that essentially looks for the nearest training image for a given input image in order to determine gaze direction for the input image [11].

Unfortunately, the previously proposed appearance-based methods share a common problem. They require eye regions to be accurately positioned in input images, which is not always easy due to the nature of low resolution images as we see in Figure 1. Even a slight *positioning error*, *i.e.*, error in the position of a cropped eye region, can degrade the accuracy of gaze estimation significantly. This important problem has not been addressed in the previous studies.

In this work, we introduce two key ideas in order to cope with the problem. One is to incorporate training images of eye regions with artificially added positioning errors. The other is to separate the factor of gaze variation from that of positioning error based on N-mode SVD (Singular Value Decomposition), which was recently introduced to the computer vision community by Vasilescu *et al.* [12]. We show that estimation of gaze direction in this framework is formulated as a bilinear problem that is then solved by alternatively minimizing a cost function with respect to gaze direction and the eye region's positioning. In order to examine how well the effect of positioning errors is removed by our method, we compared our method with an appearance-based method using PCA (Principal Component Analysis). As a result, we found that our method is able to estimate gaze directions with significantly higher accuracy than the PCA-based method.

The rest of this paper is organized as follows. In Section 2, we explain our proposed method for estimating gaze directions from low resolution images. In Section 3, we show experimental results demonstrating the merits of our proposed method. Finally, we present our concluding remarks in Section 4.

2 Proposed Method

2.1 Overview

The appearance of an eye depends not only upon gaze direction but also upon identities of subjects, poses of a head, and imaging conditions such as image resolution, response of a camera, and illumination conditions. In the present study, we focus on the problem of estimating gaze direction from low resolution images of an eye. Therefore, we do not consider appearance variations due to other factors such as identities of subjects and poses of a head. We will describe our plan for future study to deal with those factors in the Conclusions section of this paper.

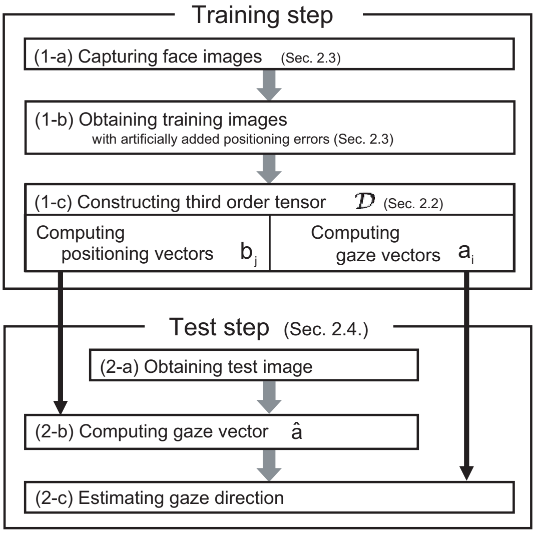

Training step
(1-a) Capturing face images (Sec. 2.3)
(1-b) Obtaining training images with artificially added positioning errors (Sec. 2.3)
(1-c) Constructing third order tensor $\mathcal{D}$ (Sec. 2.2)
Computing positioning vectors b_j
Computing gaze vectors a_i
Test step (Sec. 2.4.)
(2-a) Obtaining test image
(2-b) Computing gaze vector $\hat{a}$
(2-c) Estimating gaze direction

Fig. 2. Flowchart of our proposed method

Our proposed method consists of two steps: *the training step* and *the test step* as summarized in Figure 2. In the training step, we first capture face images with different gaze directions (1-a), and obtain an enlarged set of training images of eye regions with artificially added positioning errors (1-b). Appearance variations due to gaze direction and positioning are then modeled based on 3-mode SVD (1-c). More specifically, we construct a third order tensor from the training images, and compute a pair of two feature vectors describing gaze direction and positioning, which we call *a gaze vector* and *a positioning vector* respectively, for each training image. In the test step, we extract the gaze vector of a test image (2-b), and finally estimate the gaze direction by comparing the extracted gaze vector with those of the training images (2-c). In Section 2.2, we explain how the appearance variation of an eye for different factors is modeled by using 3-mode SVD. Then, in Sections 2.3 and 2.4, we describe the training step and the test step of our proposed method in detail.

2.2 Appearance Modeling Using 3-Mode SVD

N-mode SVD is one of the natural extensions of ordinary (2-mode) SVD to multiple modes. Here, a *mode* is a factor affecting data. For instance, identities, viewpoints, illumination conditions, facial expressions, and also image pixels can be considered as modes in face recognition [12].

Our proposed method models variations in the eye region's appearance related to three factors, *i.e.*, gaze directions, positionings, and image pixels, by using *3-mode SVD*. Here, positioning means how eye regions are cropped in input images. Let us consider a set of images of the eye region with different gaze directions and positionings. We represent those images by using a third order tensor $\boldsymbol{\mathcal{D}}$. Here, the component $\mathcal{D}_{ijk}$ $(1 \leq i \leq I, 1 \leq j \leq J, 1 \leq k \leq K)$ of the tensor is the k-th pixel value in the image with the i-th gaze direction and the j-th positioning. I, J, and K are the total numbers of different gaze directions, different positionings, and image pixels.

We can represent the third order tensor as

$$\mathcal{D}_{ijk} = \sum_{l=1}^{I} \sum_{m=1}^{J} \sum_{n=1}^{K} \mathcal{Z}_{lmn} \left(U_{\mathrm{G}}\right)_{il} \left(U_{\mathrm{POS}}\right)_{jm} \left(U_{\mathrm{PIX}}\right)_{kn}, \tag{1}$$

where $\boldsymbol{U}_{\mathrm{G}} \in \Re^{I \times I}$, $\boldsymbol{U}_{\mathrm{POS}} \in \Re^{J \times J}$, and $\boldsymbol{U}_{\mathrm{PIX}} \in \Re^{K \times K}$ are basis matrices for gaze mode, positioning mode, and pixel mode respectively, and $\mathcal{Z}_{lmn}$, which represents interaction among basis matrices, is called the *core tensor*. Basically, they correspond to two orthonormal matrices and one diagonal matrix in ordinary SVD. We also denote Eq. (1) as

$$\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{U}_{\mathrm{G}} \times_2 \boldsymbol{U}_{\mathrm{POS}} \times_3 \boldsymbol{U}_{\mathrm{PIX}}. \tag{2}$$

The basis matrix for each mode is computed as follows. First, we unfold the tensor $\boldsymbol{\mathcal{D}}$ and construct a matrix. For instance, we unfold the tensor with respect to the gaze mode G to obtain the matrix $\boldsymbol{D}_{\mathrm{G}} \in \Re^{I \times KJ}$ as $\boldsymbol{D}_{\mathrm{G}} = [\boldsymbol{F}_1 \boldsymbol{F}_2 ... \boldsymbol{F}_K]$. Here, the matrix $\boldsymbol{F}_k \in \Re^{I \times J}$ $(1 \leq k \leq K)$ is a slice of the tensor $\boldsymbol{\mathcal{D}}$ with a fixed value of k. Then, by applying SVD to the matrix $\boldsymbol{D}_{\mathrm{G}}$ as $\boldsymbol{D}_{\mathrm{G}} = \boldsymbol{U}_{\mathrm{G}} \boldsymbol{\Sigma}_{\mathrm{G}} \boldsymbol{V}_{\mathrm{G}}^{\top}$, we obtain the basis matrix $\boldsymbol{U}_{\mathrm{G}} \in \Re^{I \times I}$ of the gaze mode. The basis matrices $\boldsymbol{U}_{\mathrm{POS}}$ for the positioning mode and $\boldsymbol{U}_{\mathrm{PIX}}$ for the pixel mode are computed similarly.

The core tensor $\boldsymbol{\mathcal{Z}}$ in Eq. (2) is computed by using the tensor $\boldsymbol{\mathcal{D}}$ and basis matrices $\boldsymbol{U}_{\mathrm{G}}$, $\boldsymbol{U}_{\mathrm{POS}}$, and $\boldsymbol{U}_{\mathrm{PIX}}$ as $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{D}} \times_1 \boldsymbol{U}_{\mathrm{G}}^{\top} \times_2 \boldsymbol{U}_{\mathrm{POS}}^{\top} \times_3 \boldsymbol{U}_{\mathrm{PIX}}^{\top}$.

We define gaze vectors $\boldsymbol{a}_i \in \Re^{I}$ $(1 \leq i \leq I)$ and positioning vectors $\boldsymbol{b}_j \in \Re^{J}$ $(1 \leq j \leq J)$ as

$$[\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_I] \stackrel{\mathrm{def}}{=} \boldsymbol{U}_{\mathrm{G}}^{\top}, \quad [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_J] \stackrel{\mathrm{def}}{=} \boldsymbol{U}_{\mathrm{POS}}^{\top}. \tag{3}$$

In other words, $\boldsymbol{a}_i \in \Re^{I}$, for example, represents the i-th gaze direction in the feature space of gaze direction.

For each pair of the gaze vector $\boldsymbol{a}$ and the positioning vector $\boldsymbol{b}$, the image $\boldsymbol{d}$ of a corresponding eye region is given by using a third order tensor $\boldsymbol{\mathcal{B}}$

$$\mathcal{B}_{ijk} \stackrel{\mathrm{def}}{=} \sum_{l=1}^{K} \mathcal{Z}_{ijl} \left(U_{\mathrm{PIX}}\right)_{kl} \tag{4}$$

as

$$d_k = \sum_{i=1}^{I}\sum_{j=1}^{J} \mathcal{B}_{ijk} a_i b_j = \sum_{i=1}^{I}\sum_{j=1}^{J} B_{k(ij)} a_i b_j. \tag{5}$$

Here, we represent the tensor $\mathcal{B}$ in the matrix form B. $B_{k(ij)}$ is the $(k, I\times(j-1)+i)$ component of the matrix B_{PIX}, which is obtained by unfolding $\boldsymbol{\mathcal{B}}$ with respect to the pixel mode.

2.3 Training Step

In the training step, variations in appearance of eye regions are learned from training images that are down-sampled from high resolution images of eyes. This is done because, unlike test images, high resolution training images are easily available, and, more importantly, the position of an eye can be found accurately in high resolution images by using existing techniques. For the present study, we used our feature-based face tracker [8] for finding eye corners.

We first capture a set of high resolution images of an eye with different but known gaze directions. Then, eye region images without positioning errors are obtained by cropping rectangular regions from down-sampled images by using the positions of eye corners. In addition, eye region images with artificially added positioning errors are obtained by moving the positions of eye corners.

After a set of training images of eye regions is created, we construct the third order tensor $\boldsymbol{\mathcal{D}}$ from the training images and obtain the gaze vectors $\boldsymbol{a}_i$ $(1\leq i\leq I)$ and the positioning vectors $\boldsymbol{b}_j$ $(1\leq j\leq J)$ from Eq. (3) as described in Section 2.2. Additionally, we prepare the matrix $B_{k(ij)}$ from Eq. (4).

2.4 Test Step

For each test image, an eye region is found first. It should be noted that, unlike in the training step, the image resolution of the eye region is not necessarily high. However, some existing techniques for facial component detection, *e.g.*, the AdaBoost algorithm [3] and the Gabor-like feature filtering scheme [16], can find eye regions even in low resolution test images.

After an eye region is found, the gaze vector is computed for the eye region. Then, the gaze direction for the test image is determined from the gaze vector. We will explain each of the steps in this section.

(1) Extraction of Gaze Vectors. In order to extract feature vectors from test images, two methods based on projections have been proposed. The first one proposed by Vasilescu [12] for face recognition projects a test image into the feature space of face identity by using a set of matrices. The method uses one matrix per each combination of indices except for that corresponding to identity mode, and thus only yields a set of candidates for the correct feature vector.

The second method recently proposed by Vasilescu [13] uses a single matrix independent of specific values of modes. This method is more elegant than the first one in the respect that it can simultaneously extract unique feature vectors of all modes from a test image. However, the method is not applicable to our purpose of extracting feature vectors from low resolution images. The method assumes that the number of pixels is larger than the product of the number of indices in each mode. This assumption, which requires $K \geq IJ$ in our case, is not satisfied.

Accordingly, we introduce an algorithm for extracting feature vectors from low resolution images in the context of 3-mode SVD. Let us consider a test image $\hat{\boldsymbol{d}}$, and an image constructed from a gaze vector $\boldsymbol{a}'$ and a positioning vector $\boldsymbol{b}'$ through Eq. (5). Then, we define a cost function $f(\boldsymbol{a}', \boldsymbol{b}')$ by $f(\boldsymbol{a}', \boldsymbol{b}') \stackrel{\text{def}}{=} \sum_{k=1}^{K} \left(\hat{d}_k - \sum_{i=1}^{I} \sum_{j=1}^{J} \left(B_{k(ij)} a'_i b'_j \right) \right)^2$, and estimate the feature vectors of the test image $(\hat{\boldsymbol{a}}, \hat{\boldsymbol{b}})$ by minimizing the cost function as $(\hat{\boldsymbol{a}}, \hat{\boldsymbol{b}}) = \arg\min_{\boldsymbol{a}' \in \Re^I, \boldsymbol{b}' \in \Re^J} f(\boldsymbol{a}', \boldsymbol{b}')$.

This cost function is bilinear, that is, it is linear with respect to one variable $\boldsymbol{a}'$ when the other variable $\boldsymbol{b}'$ is fixed, and vice versa. Therefore, we can directly and uniquely extract the feature vectors by alternatively minimizing the bilinear cost function in a similar manner to Shum [9]. Our proposed method thus relaxes the requirement with respect to the number of pixels from $K \geq IJ$ to $K \geq (I + J)$. In other words, for test images with a fixed image resolution, our method can use a wider variety of training images than the previously proposed method [13].

More specifically, the solutions of linear problems with respect to one variable $\partial f / \partial a'_i = 0 \ \ (1 \leq i \leq I)$ and $\partial f / \partial b'_j = 0 \ \ (1 \leq j \leq J)$ result in

$$\boldsymbol{a}' = \boldsymbol{M}^{+} \hat{\boldsymbol{d}}, \quad \boldsymbol{b}' = \boldsymbol{N}^{+} \hat{\boldsymbol{d}}, \quad (M)_{ki} \stackrel{\text{def}}{=} \sum_{j=1}^{J} B_{k(ij)} b'_j, \quad (N)_{kj} \stackrel{\text{def}}{=} \sum_{i=1}^{I} B_{k(ij)} a'_i, \quad (6)$$

where the matrix $\boldsymbol{M}^{+}$ is the pseudoinverse of $\boldsymbol{M}$. Therefore, we can assign, for example, the initial value of $\boldsymbol{b}'$ to $\boldsymbol{b}'^{(0)}$, and alternatively update the feature vectors according to Eq. (6) until they converge. Actually, we terminate the iteration when $\Delta f(n) \stackrel{\text{def}}{=} f(\boldsymbol{a}'^{(n)}, \boldsymbol{b}'^{(n)}) - f(\boldsymbol{a}'^{(n-1)}, \boldsymbol{b}'^{(n-1)})$ is less than the predefined threshold. Here, we denote the feature vectors at the n-th iteration as $\boldsymbol{a}'^{(n)}$ and $\boldsymbol{b}'^{(n)}$. In this way, the gaze vector is determined up to an unknown scale factor. Therefore, we normalize $\boldsymbol{a}'$ by using the L_2 norm to obtain the gaze vector $\hat{\boldsymbol{a}}$ for the given test image.

In the current implementation, we choose the initial value of $\boldsymbol{b}'$ according to $(\boldsymbol{a}'^{(0)}, \boldsymbol{b}'^{(0)}) = \arg\min_{\boldsymbol{a}' \in \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_I\}, \boldsymbol{b}' \in \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_J\}} f(\boldsymbol{a}', \boldsymbol{b}')$. Namely, we search for the combination of the gaze and positioning vectors of training images that yields the image most similar to the test image in the least-square sense. Though the combination of the initial value might provide local minima of the cost function, our experimental results imply that the local minima do not affect the results seriously.

(2) Estimation of Gaze Direction. Finally, we determine the gaze direction for the given test image by using the obtained gaze vector. We find three gaze directions of the training images nearest to that of the test image, and calculate the gaze direction of the test image by interpolating them. We do this because we need at least three gaze directions to represent an arbitrary gaze direction by interpolation. First, we find the index of the gaze vector of the training image that is the closest to the obtained gaze vector as $i\,(1) = \arg\min_{i \in \{1,2,\ldots,I\}} |\hat{\boldsymbol{a}} - \boldsymbol{a}_i|^2$. Similarly, we find the indices $i(2)$ and $i(3)$ of the second and third closest gaze vectors. Then we determine the gaze direction by interpolating the three gaze directions such that the interpolated gaze vector becomes the closest to the obtained gaze vector of the test image. This is done by choosing the three weights w_p $(p = 1, 2, 3)$ that minimize $|\hat{\boldsymbol{a}} - \sum_{p=1}^{3} w_p \boldsymbol{a}_{i(p)}|^2$ subject to $0 \leq w_p \leq 1$ and $\sum_{p=1}^{3} w_p = 1$. Then the gaze direction $\boldsymbol{g}$ is given as $\boldsymbol{g} = \sum_{p=1}^{3} w_p \boldsymbol{g}(p)$, where $\boldsymbol{g}(p)$ is the gaze direction of $i(p)$.

3 Experiments

3.1 Eye Images for Experiments

(1) Imaging Conditions. We captured facial images of five individuals, and estimated gaze direction of each subject using our proposed method. In our experiments, we evaluated the accuracy of gaze estimation for each subject separately, *i.e.*, using training and test images of the same subject for gaze estimation. This was done because we do not deal with appearance variation due to different identities of subjects. To quantitatively evaluate the accuracy of our method, we captured images while subjects stared at targets appearing on an 18-inch SXGA monitor placed at a distance of 50cm from the subject's face. Since we calibrated the relative position of the monitor in advance of capturing images, we could calculate gaze direction corresponding to a 2D position on a monitor when a user was looking at the position.

Figure 3 shows the target positions displayed on the monitor: circles for training and crosses for test. Twenty training images and 32 test images were taken for each subject. Since we do not consider face pose change in this study, we asked subjects not to move their heads while images were being taken. Each subject was asked to move a mouse pointer to a randomly appearing target and press the mouse button while the pointer was placed on the target to capture a face image of the subject staring at the target.

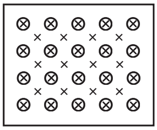


Fig. 3. A layout of targets displayed on the surface on an LCD monitor: circles for training and crosses for test

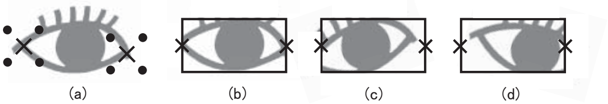


Fig. 4. (a) A schematic illustration of the correct corners of an eye (crosses) and the points used for artificially representing incorrect positioning (dots). (b) An illustration of an eye image cropped based on the correct positioning. (c) and (d) show those cropped with the incorrect positionings. We cropped eye regions so that the feature points in Figure 4 (a) are aligned to the crosses on both sides.

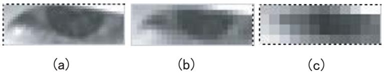


Fig. 5. Example images of an eye with (a) 16 × 48, (b) 8 × 24, and (c) 4 × 12 pixels

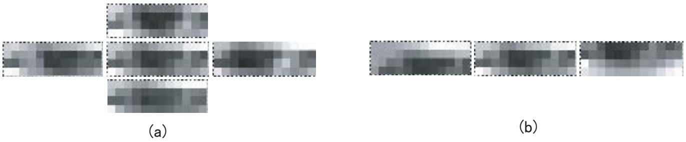


Fig. 6. (a) Images cropped based on the correct positioning with various gaze directions. (b) Images cropped based on the various positionings with a fixed gaze direction.

(2) Cropping Eye Regions. After all face images were captured, we prepared eye region images for both training and testing. Note that we used down-sampled images for test images instead of images captured at low resolution in our experiments. This was necessary for evaluating the accuracy of our method quantitatively. The use of down-sampled test images enables us to investigate (i) how the accuracy of gaze estimation is affected by inaccurate positioning of eye regions, and (ii) how the estimation accuracy changes depending on the resolution of test images.

Eye regions were cropped from down-sampled images by using positions of eye corners found by our feature-based face tracker [8] and additional positions that were shifted diagonally from those true eye corners by one step[1]. Figure 4 (a) shows a schematic illustration of true eye corners and additional positions with artificially added positioning errors. In this way, we prepared 25 ($= 5 \times 5$) eye region images per gaze direction. All the images were aligned by affine transformations as illustrated in Figure 4 (b), (c), and (d).

For testing our method with different image resolutions, we used images with 16×48, 8×24, and 4×12 pixels as shown in Figure 5. As we see in those figures, it is impossible to localize geometric features such as the iris and cornea of an eye if image resolution is too low.

We show examples of eye regions for different gaze directions in Figure 6 (a). Those regions were cropped by using correct positions of eye corners. We also show eye regions for the same gaze direction but cropped with positioning errors in Figure 6 (b). From these examples, we see it is not trivial to estimate gaze directions from low resolution images without being affected by poor positioning accuracy.

3.2 Experimental Results

We quantitatively evaluated the performance of our proposed method, and compared the performance with that of a method based on conventional PCA from three aspects:

[1] We define one step as 4 pixels, 2 pixels, or 1 pixel for each eye image with 16 × 48, 8 × 24, or 4 × 12 pixels respectively.

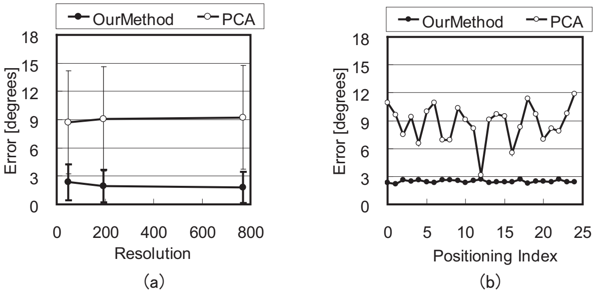


Fig. 7. (a) Errors of gaze estimation against image resolution. (b) Gaze estimation error for each positioning. Index 12 indicates the correct positioning.

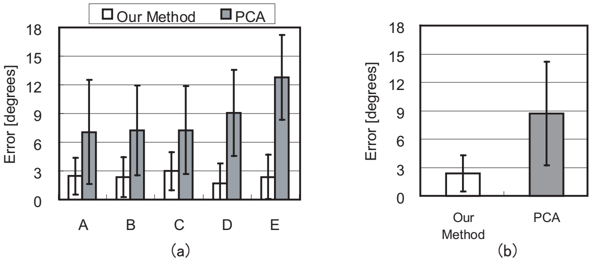


Fig. 8. (a) Gaze estimation errors for each individual and (b) gaze estimation error averaged over all individuals

how the estimation accuracy changes depending on image resolution, positionings, and individuals. The PCA-based method does not treat variations due to changes in gaze direction and position separately, and projects a test image into the feature space defined by the principal axes computed by using all training images with various gaze directions and positionings. The feature vector of one gaze direction is defined by the average of feature vectors computed for images with the same gaze direction but various positionings. In order to alleviate any bias due to brightness variation among images, we normalized training and test images for both our method and the PCA-based method so that pixel values in each image have zero mean and unit variance. We used 3-dimensional feature space for both our method and the PCA-based method to estimate gaze direction.

Estimation Error Against Image Resolution. First, we show errors of gaze estimation against image resolutions in Figure 7 (a). The horizontal axis indicates the number of pixels in the eye images ($4 \times 12 = 48$, $8 \times 24 = 192$, and $16 \times 48 = 768$), and the vertical axis represents the average and standard deviation of errors over five subjects. This figure shows that the accuracy of our proposed method is higher than that of the

PCA- based method. Hereafter, we show results for eye images with the lowest resolution, that is, with 4×12 pixels.

Estimation Error for Each Positioning. Second, we show errors for test images with various positionings in Figure 7 (b). The horizontal axis indicates the index number j for the positionings of the eye regions. Here, $j = 12$ corresponds to the correct positioning. The vertical axis represents the averaged error of five subjects.

Comparing the error at $j = 12$ with those for other indices, it is clear that the PCA-based method is sensitive to positioning errors. On the other hand, the errors of our proposed method are almost the same for all positionings. Therefore, we can conclude that our method is robust against positioning errors.

Estimation Error for Each Individual. Finally, we show the estimation error of five subjects A, B, C, D, and E in Figure 8 (a), and the error averaged over the five subjects in (b). This figure shows that the performance of our proposed method is better than that of the PCA-based method for all subjects.

Note that the averaged error—2.4 degrees in Figure 8 (b)—is less than half of the sampling distance of the training images—6.4 degrees, the distance between the nearest two circles in Figure 3. The experimental results demonstrate that our bilinear model of two factors, gaze direction and eye region's positioning, can accurately represent the appearance variations resulting from the different gaze directions and positionings.

4 Conclusions

In this study, we proposed a new appearance-based method for gaze estimation from low resolution images, and demonstrated the merit of our proposed method via a number of experiments. One of the key challenges for gaze estimation from low resolution images is that eye regions cannot be found accurately due to limited image resolution, which results in inaccurate estimation of gaze directions. Unlike previously proposed methods, our method is able to estimate gaze directions accurately even when eye regions are found inaccurately in input images.

In order to realize gaze estimation that is insensitive to positioning errors, our method models appearance variation of eye regions due to not only changes in gaze direction but also changes in positioning of eye regions. This is done by incorporating training images of eye regions with artificially added positioning errors, and separating the factor of gaze variation from that of positioning error with a method based on N-mode SVD. In addition, we showed how the problem of gaze estimation can be formulated as a bilinear problem which is solved by alternatively minimizing its cost function with respect to gaze direction and localization of eye regions.

In the present study, we focused on the problem caused by inaccurate positioning of eye regions in low resolution images. Therefore, we did not consider appearance variations due to other factors such as subject identities and head poses. For our future work, we are planning to extend our method to deal with those factors by incorporating additional modes in the N-mode SVD framework.

References

1. S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. In *Advances in Neural Information Processing Systems*, pages 753–760, 1993.
2. D. Beymer and M. Flickner. Eye gaze tracking using an active stereo head. In *Proc. IEEE CVPR*, pages 451–458, 2003.
3. D. Cristinacce and T. Cootes. Facial feature detection using adaboost with shape constraints. In *Proc. British Machine Vision Conference*, pages 231–240, 2003.
4. T. Hutchinson, K. White JR, W. Martin, K. Reichert, and L. Frey. Human-computer interaction using eye-gaze input. *IEEE Trans. on Systems, Man, and Cybernetics*, 19(6):1527 – 1534, 1989.
5. T. Ishikawa, S. Baker, I. Matthews, and T. Kanade. Passive driver gaze tracking with active appearance models. In *Proc. Intelligent Transportation Systems*, October 2004.
6. Y. Matsumoto and A. Zelinsky. An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. In *Proc. IEEE FG*, pages 499–505, 2000.
7. T. Ohno and N. Mukawa. A free-head, simple calibration, gaze tracking system that enables gaze-based interaction. In *Proc. Eye Tracking Research and Application symposium*, pages 115–122, 2004.
8. K. Oka, Y. Sato, Y. Nakanishi, and H. Koike. Head pose estimation system based on particle filtering with adaptive diffusion control. In *IAPR Conf. Machine Vision Applications (MVA 2005)*, pages 586–589, May 2005.
9. H.-Y. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Trans. PAMI*, 17(9):854–867, 1995.
10. R. Stiefelhagen, J. Yang, and A. Waibel. Tracking eyes and monitoring eye gaze. In *Proc. Workshop on Perceptual User Interfaces*, Banff, Canada, October 1997.
11. K.-H. Tan, D. Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 191–195, 2002.
12. M. A. O. Vasilescu and D. Terzopoulos. Multilinear image analysis for facial recognition. In *Proc. ICPR*, pages 511–514, 2002.
13. M. A. O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. In *Proc. IEEE CVPR*, pages 547–553, 2005.
14. J.-G. Wang, E. Sung, and R. Venkateswarlu. Eye gaze estimation from a single image of one eye. In *Proc. IEEE ICCV*, pages 136–143, 2003.
15. L.-Q. Xu, D. Machin, and P. Sheppard. A novel approach to real-time non-intrusive gaze finding. In *Proc. British Machine Vision Conference*, 1998.
16. T. Yao, H. Li, G. Liu, X. Ye, W. Gu, and Y. Jin. A fast and robust face location and feature extraction system. In *Proc. IEEE ICIP*, pages 157–160, 2002.
17. D. Yoo and M. Chung. Non-intrusive eye gaze estimation without knowledge of eye pose. In *Proc. IEEE FG*, pages 785–790, 2004.
18. Z. Zhu and Q. Ji. Eye gaze tracking under natural head movements. In *Proc. IEEE CVPR*, pages 918–923, 2005.

Image Similarity Comparison Using Dual-Tree Wavelet Transform

Mong-Shu Lee, Li-yu Liu, and Fu-Sen Lin

Department of Computer Science
National Taiwan Ocean University
Keelung, Taiwan R.O.C
{mslee, M93570030, fslin}@mail.ntou.edu.tw

Abstract. An image similarity comparison method for images with minor distortions is introduced in this paper. The proposed image similarity metrics is based on a new method to measure structure similarity for image quality comparisons. We make use of the fact that Dual-Tree wavelet Transform (DTWT) can provide direction selectivity and keep the structure features between the original and images with minor distortions. Despite the simplicity of our method, our experimental results demonstrate the effectiveness of the proposed method.

Keywords: Image similarity, Wavelet transform.

1 Introduction

Over the last decade or so, the discrete wavelet transform (DWT) has been successfully used in the signal processing field for a variety of reasons: The wavelet transform is fast, local in the time and frequency domain, and it provides multi-resolution analysis of real world signals and images. Unfortunately, the DWT also has some disadvantages that undermine its broader use in signal and image processing applications. First, it is shift sensitive, and small shifts in the input signal can cause abrupt variations in the distribution of energy between wavelet coefficients at different scales. Second, the DWT coefficients have poor directional selectivity. To overcome these problems, some other wavelet transforms have been studied recently. For example, over-complete wavelet transform, discard all down-sampling in DWT to achieve shift invariance. Unfortunately, this method incurs great computational cost, and the issue of poor directional selectivity remains unsolved. Several authors[2], [6] have proposed that in a formulation where two dyadic wavelet bases form a Hilbert transform pair, the DWT can provide the answer to some of the aforementioned limitations. The Kingsburg's dual-tree wavelet transform (DTWT) generates complex coefficients by using dual tree filters to obtain their real and imaginary parts.

Image similarity measurement is one of the important issues in information processing, and a major challenge for computer science. For example, finding the appropriate similarity measures between extracted features is the key task for content-based image retrieval.

A simple but inefficient way to evaluate the similarity between two images is to use a simple distance measure, such as the mean square error (MSE), which is easy to

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 189 – 197, 2006.

calculate and is mathematically convenient. However, it does not provide a consistent relationship with the quality perceived by the human visual system (HVS). Recently, Wang *et al.* [8] have developed a measure of structure similarity (SSIM) for image quality assessment. The SSIM metrics models perception implicitly by taking into accounts high-level HVS characteristics. They showed that the simple SSIM algorithm provides excellent image quality prediction performance for various distorted images. The proposed approach for similar images comparison is motivated by the fact that the DTWT provides good directional selectivity for extracting the global features of images, and therefore they are directly related to structure similarity in the image match. In this paper, our goal is to extend the current SSIM method to the dual-tree wavelet transform domain, and make it become an image similarity metrics, called dual-tree wavelet transform SSIM (DTWT-SSIM). We model the distorted images by the familiar affine transformations and show that the introduced DTWT-SSIM index is stable under the affine transformations. Our experimental results illustrate that the proposed image similarity measure yields a significantly superior identification rate than the MSE and SSIM methods when the distortion of translation, scaling and rotation is small.

2 Dual-Tree Wavelet Transform

As shown in Fig. 1, in the one-dimensional DTWT, two real wavelet trees are used, each capable of perfect reconstruction. One tree generates the real part of the transform and the other one is used in generating the complex part. In Fig. 1, $h_0(n)$ and $h_1(n)$ are the low-pass and high-pass filters of a Quadrature Mirror Filter (QMF) pair in the analysis branch. For the complex part, $\{g_0(n), g_1(n)\}$ is another QMF pair in the analysis branch. All filter pairs discussed here are orthogonal and real-valued. Each tree produces a valid set of real DWT coefficients u_i and v_i, and together they form the complex coefficients $d_i = u_i + jv_i$. It has been shown [7] that if filters in both trees can be made to be offset by a half sample, then the two wavelets satisfy the Hilbert transform pair condition.

A separable two-dimensional DWT can be computed efficiently in discrete space by applying the associated one-dimensional filter to each column of the image, and then applying the filter to each of the resultant coefficients. Therefore a normal two-dimensional DWT produces four band-pass sub-images at each level, corresponding to low-low, low-high, high-low, and high-high filtering. As with one-dimensional DWT, the low-low parts coefficients represent the smooth version of the original function. However, the other three sub-bands wavelet coefficients of two-dimensional DWT capture features along lines at angles of $\{0^\circ, 90^\circ, 45^\circ\}$. To overcome the drawbacks of DWT, Kingsbury [2] have developed the DTWT, which allows perfect reconstruction while still providing shift invariance and directional selectivity. The DTWT transform has the ability to differentiate positive and negative frequencies, and it produces six band-pass sub-images of wavelet coefficients at each level, all of which are strongly oriented at angles of ±15°, ±45°, and ±75°. The DTWT expansion of an image $f(x)$ is given by

$$f(x) = \sum_k c_\phi(j_0,k)\phi_{j_0,k}(x) + \sum_i \sum_{j\ge j_0} \sum_k d_\psi(j,k)\psi^i_{j,k}(x), \text{ where } i = \pm 15^\circ, \pm 45^\circ, \text{ and } \pm 75^\circ.$$

$c_\phi(j_0,k)$ and $d_\psi(j,k)$ are the scaling and wavelet coefficients of the DTWT, using dual-tree scaling functions $\phi_{j_0,k}$ and wavelet functions $\psi^i_{j,k}$, respectively. For the sake of simplicity of notation, from here on we will denote the wavelet coefficients $d_\psi(j,k)$ of an image $f(x)$ as d_x later.

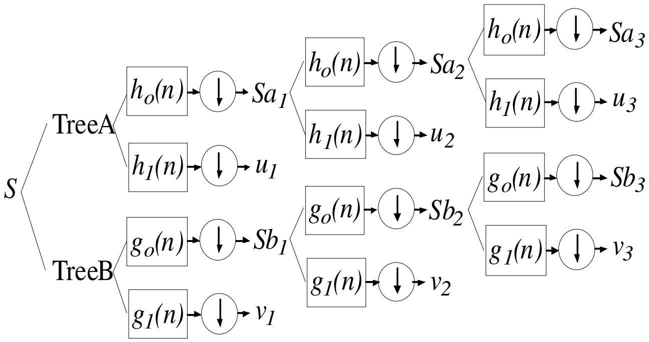
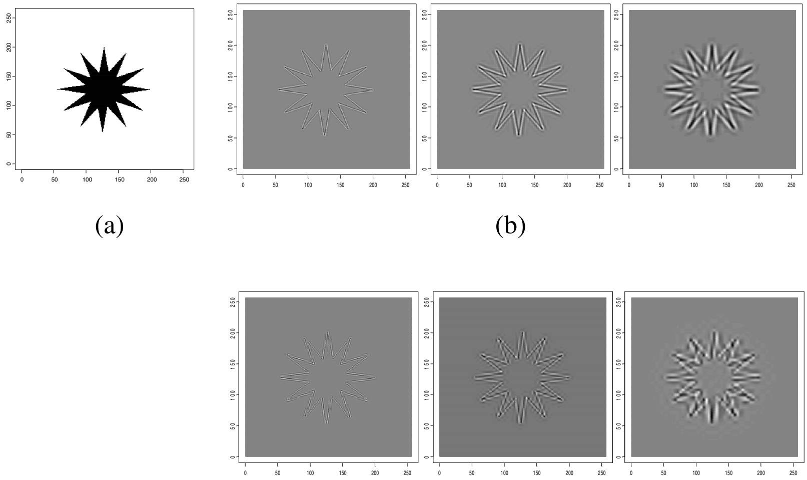


Fig. 1. Kingsbury's Dual-Tree Wavelet Transform with three levels of decomposition



(a)

(b)

(c)

Fig. 2. (a) The star image. (b) The reconstructed images, from left to right, at levels 1, 2, and 3 for the DTWT. (bc) The reconstructed images, from left to right, at levels 1, 2, and 3 for the DWT.

For the sake of comparison, the reconstructed images, from left to right, at levels 1, 2, and 3 for the DWT are shown below the DTWT in Fig. 2. Clearly, the presence of directional selectivity in the DTWT shows its ability to extract the structure or connectedness of natural images.

3 Image Similarity

3.1 DTWT-SSIM Index

This application of the DTWT for image similarity assessment is inspired by the success of the spatial domain structural similarity (SSIM) index algorithm [8]. The principle of the structural approach is that the human visual system is highly adapted and capable of extracting structural information (the structure of the objects) from a visual scene. As a result, a measure of structure similarity should be a good approximation of image similarity. In the spatial domain, the SSIM index that quantizes the luminance, contrast and structure changes between two image patches $\mathbf{x}=\{x_i \mid i=1,...M\}$ and $\mathbf{y}=\{y_i \mid i=1,\ ...M\}$ is defined as [8]

$$S(x,y)=\frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}, \tag{1}$$

where C_1 and C_2 are two small positive constants; $\mu_x=\frac{1}{M}\sum_{i=1}^{M}x_i$, $\sigma_x^2=\frac{1}{M}\sum_{i=1}^{M}(x_i-\mu_x)^2$, and $\sigma_{xy}=\frac{1}{M}\sum_{i=1}^{M}(x_i-\mu_x)(y_i-\mu_y)$.

Roughly speaking μ_x and σ_x can be regarded as estimates of the luminance and contrast of x, while σ_{xy} measures the tendency of x and y to vary together. It can be shown that the maximum SSIM index value equals 1 if and only if **x** and **y** are identical.

A major drawback of the spatial domain SSIM algorithm is that it is highly sensitive to translation, scaling and rotation of images. It must be remembered that the DTWT is approximately shifted invariant and directionally selective. So, hopefully the similar global structure of minor distorted images can be extracted by comparing their DTWT coefficients. Therefore we attempt to extend the current SSIM approach to the dual tree wavelet transform domain and make it insensitive to small "non-structure" geometric distortions caused by the image capturing process, rather than by the changes of the structures of the objects in the visual scene.

In the dual tree wavelet transform domain, let us suppose that $d_x=\{d_{x,i}^{j} \mid i=1,2,...,N \text{ and } j=1,...,6\}$ and $d_y=\{d_{y,i}^{j} \mid i=1,2,...,N \text{ and } j=1,...,6\}$ are two sets of the DTWT wavelet coefficients extracted from one of the decomposition levels at the six sub-bands of the images x and y. We replace the μ_x and μ_y in the Eq. (1) by summing all the six sub-bands of DTWT coefficients.

The concept of total sum is precisely equivalent to the average when the numerator and the denominator in Eq. (1) have the same divisor. Now the spatial domain SSIM index is naturally extended to a DTWT domain SSIM as follows.

$$DTWT - SSIM(x,y) = \frac{(2\mu_{d_x}\mu_{d_y} + K_1)(2\sigma_{d_x d_y} + K_2)}{(\mu_{d_x}^2 + \mu_{d_y}^2 + K_1)(\sigma_{d_x}^2 + \sigma_{d_y}^2 + K_2)}$$

$$= \frac{\left(2\mu_{|d^j_{x,i}|}\mu_{|d^j_{y,i}|} + K_1\right)\left(2\sum_{j=1}^{6}\sum_{i=1}^{N}(|\,d^j_{x,i}\,| - \mu_{|d^j_{x,i}|})(|\,d^j_{y,i}\,| - \mu_{|d^j_{y,i}|}) + K_2\right)}{\left((\mu_{|d^j_{x,i}|})^2 + (\mu_{|d^j_{y,i}|})^2 + K_1\right)\left(\sum_{j=1}^{6}\sum_{i=1}^{N}(|\,d^j_{x,i}\,| - \mu_{|d^j_{x,i}|})^2 + \sum_{j=1}^{6}\sum_{i=1}^{N}(|\,d^j_{y,i}\,| - \mu_{|d^j_{y,i}|})^2 + K_2\right)} \qquad (2)$$

Here $|\,d^j_{x,i}\,|$ denotes the modulus (absolute value) of the complex numbers $d^j_{x,i}$, and K_1 and K_2 are small positive constants to avoid instability when the denominator is very close to zero.

3.2 Sensitivity Measure

The affine transformation is a convenient way to describe geometric distortion in many imaging system. A planar affine transformation is equivalent to the composed effect of three linear transformations, translation, rotation and scaling. Now we can describe the image translation, rotation and scaling operation by matrices and coordinate system as follow. The general affine transformation is commonly written in the familiar x,y-notation for coordinates in the plane.

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = A\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + B,$$

where $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$, $\begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$ represents the pixel intensity located at position $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$ in the reference and altered version, respectively, and matrix A and vector B specify the desired operation. For example, by defining $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, this transformation, can carry out pure translation. Pure rotation uses the $A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Similarly, the pure scaling operation is $A = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

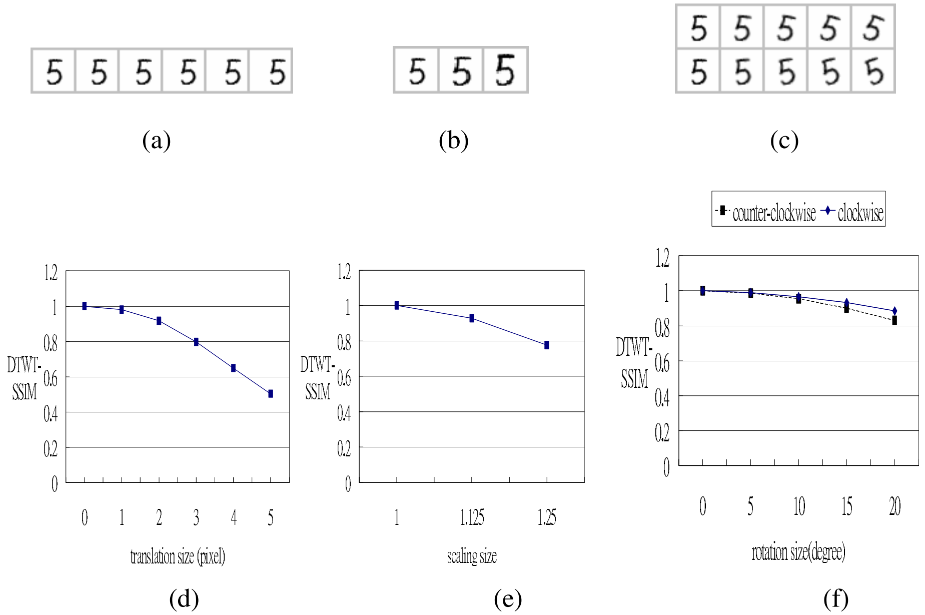
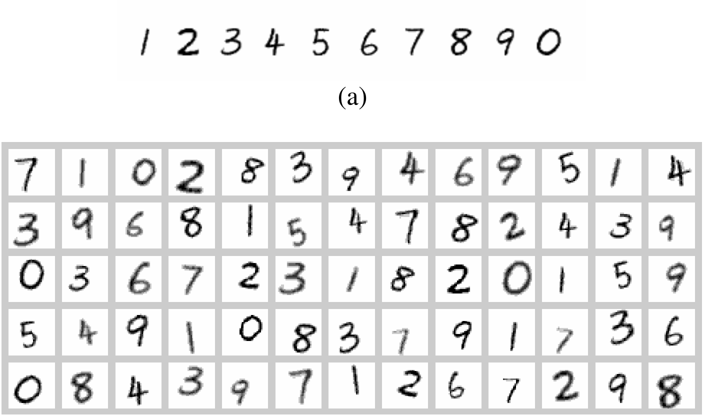


Fig. 3. DTWT-SSIM results under different value of translation, scaling (zooming) and rotation

We also know that the condition number $\kappa(A)$ quantifies the sensitivity of a specified transformation problem. Define the condition number $\kappa(A)$ by $\kappa(A) = \| A \|_\infty \| A^{-1} \|_\infty$, where A is a $n \times n$ matrix and $\| A \|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} | a_{ij} |$. For a non-singular matrix, $\kappa(A) = \| A \|_\infty \| A^{-1} \|_\infty \ge \ \| A \cdot A^{-1} \|_\infty = \| I \|_\infty = 1$ .In general matrices with small condition number, $\kappa(A) \cong 1$, are said to be well-conditioned. It is obvious to see that all the pure affine transformation matrices are well-conditioned. Furthermore, we have that the composition matrix of these well-conditioned affine transformations still satisfies $\kappa(A) \cong 1$. Let A_1, A_2 be any of the pure affine transformation, by using $\kappa(A_1 A_2) \le \kappa(A_1)\kappa(A_2)$, we conclude that the composition of any two of these affine transformations also satisfies $\kappa(A_1 A_2) \cong 1$ .Therefore the affine transformation is insensitive to small distortions. An example of showing the stability of DTWT-SSIM index under different affine transformation is depicted in Fig. 3. The original digit image “5” is shown in the leftmost of Fig. 3(a), 3(b), and 3(c), then with the different numbers of distorted images, including translation, scaling (zooming) and rotation. The DTWT-SSIM values under different affine transformations are shown in Fig. 3(d), 3((e), and 3(f), respectively. We can see that the DTWT-SSIM index between the original image and the distorted image steadily decreases as the distortion increases. Besides that, the DTWT-SSIM has various decreasing rates corresponding to the translation, scaling and rotation transformation as the distortion of the three affine transformations increases.

4 Test Results

To demonstrate the efficiency of the DTWT domain SSIM measure task, we conduct a handwritten digit matching experiment using the MNIST handwritten digit data base. In the case of handwritten digit recognition, this means that digits of each test class contain position displacement, size change, slight rotations distortion or changes in line thickness. As shown in Fig. 4(a), we have ten standard digit templates (MNIST samples) with each size of 32x32 pixels.



(a)

(b)

Fig. 4. (a) Standard digit templates. (b) Subset of test images (randomly selected from 4860 images).

To evaluate the DTWT-SSIM measure for comparing images, we apply the Q-shift version of the DTWT with three levels of decomposition to the two given image being compared. It is well known that the amount of energy increases toward the low frequency sub-bands after decomposing the original image into several sub-bands with general wavelet transforms [4]. Therefore we calculate the DTWT-SSIM index using Eq. (2) with only the lowest sub-band coefficients. Also we compute the DTWT-SSIM index using the original image size 32x32 because the window size from 4x4 to 32x32 is feasible to compute the distortions.

To model some possible instances of a category, it must be present in the prototype set. So, we create a total of 4860 artificial images by combing shifting to the right or left (three pixels), scaling (10%), rotating (up to 20 degrees clockwise or counter-clockwise), and blurring the standard templates (see Fig. 4(b)).

MSE, SSIM, and DTWT- SSIM are used as the matching criterion in the following matching procedure. We first choose one image from the 4860 distorted images as the test image, and then a 3-level DTWT is applied to decompose the test image. The next step is to compute the similarity index between the test image and the ten standard digit templates. The test digit image is then "identified" as belonging to the category that corresponds to the highest similarity score among the ten standard templates. If the resulting test digit image is in the same category as it should be, then we say it matched, otherwise we say it is unmatched (Fig. 5.).

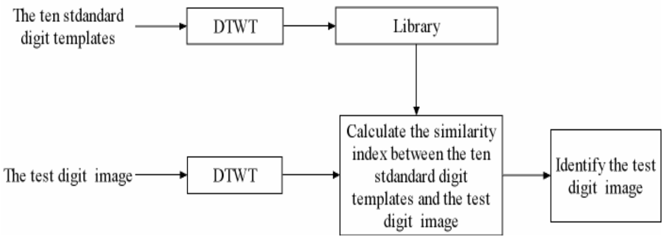

The ten stdandard digit templates
DTWT
Library
The test digit image
DTWT
Calculate the similarity index between the ten stdandard digit templates and the test digit image
Identify the test digit image

Fig. 5. Block diagram of the proposed matching procedure

Table 1. Correct identification rate using different similarity metrics (%)

digit	MSE	SSIM	DTWT-SSIM
1	68.11%	34.36%	94.03%
2	19.55%	36.83%	92.18%
3	12.96%	20.58%	90.95%
4	19.96%	34.57%	87.24%
5	13.58%	21.19%	95.27%
6	19.14%	17.90%	94.44%
7	22.63%	18.93%	78.19%
8	8.02%	16.67%	82.30%
9	9.88%	15.23%	86.01%
0	17.70%	38.27%	93.00%
Average	21.15%	25.45%	89.36%

The identification performance is significantly different when different similarity measures are employed. The resulting correct identification rates are shown in Table 1. The identification match rate of the MSE and the spatial domain SSIM are low, as expected, since both measures are sensitive to translation, scaling and rotation of images. Table 1 shows the correct identification rate of the MSE and the SIM is as low as 25%. By contrast, the correct identification rate of the DTWT domain SSIM gives the best result, 89%.

5 Conclusion

The proposed DTWT domain SSIM image similarity index method is easy to implement. Even though hardly any pre-processing or training is required, the performance result of the presented method is considerable better than those of the traditional MSE or SSIM method. It is our conclusion that the main reasons for this success are first due to the fact that the dual-tree wavelet transform provides good directional selectivity in six orientations at dyadic scales. Secondly, the image translation, rotation and scaling transformations are stable to the small perturbations. They all contribute the ability to substantiate the image structure similarity index.

This introduced method is still in its infancy. We are working on developing it into a more systematic approach that can potentially be employed in a much broader range of applications, such as face recognition, or content-based image retrieval. Both of these two research areas put emphasis on finding similar geometric structure of objects or scenes and thus, it is suitable for the proposed DTWT-SSIM to gain exploitations.

Acknowledgments

This research was supported by the National Science Council of R. O. C. Taiwan. (NSC 93-2115-M-019-002 and NSC 95-2115-M-019-002).

References

[1] N. G. Kingsbury, "Image Processing with Complex Wavelets", Phil. Trans. R. Soc. London. A, Sept. 1999.

[2] N. G. Kingsbury, “Complex wavelets for shift invariant analysis and filtering of signals,” *Appl. Comput. Harmon. Anal.*, vol. 10, no 3, pp. 234-253, May 2001.

[3] N. G. Kingsbury, “A dual-tree complex wavelet transform with improved orthogonality and symmetry properties,” *in Proc. IEEE Int. Conf. Image Processing,* Vancouver, BC, Canada, Sept. 10-13, 2000.

[4] O. J. Kwon and R. Chellappa, “Region adaptive subband image coding”, IEEE Transactions on Image Processing. Volume 7, Issue 5, May 1998 pp. 632 – 648.

[5] M. J. T. Smith and T. P. Barnwell III, “Exact reconstruction techniques for tree-structured subband coders,” *IEEE, Acoustics, Speech, and Signal Processing,* vol. 34, pp.431-441, June 1986.

[6] I. W. Selesnick. “The design of approximate Hilbert transform pairs of wavelet bases,” *IEEE Trans. on Signal Processing,* vol. 50, pp.1144-1152, Mar 2002

[7] I. W. Selesnick. “Hilbert transform pairs of wavelet bases,” *IEEE Signal Processing Lett.*, vol. 8, pp. 170-173, June 2001.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error Visibility to structural similarity,” *IEEE Trans, Image Processing,* vol. 13, pp. 600-612, Apr. 2004.

[9] Zhou Wang and E. P. Simoncelli, “Translation Insensitive Image Similarity in Complex Wavelet Domain”, *IEEE, Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05).* vol. 2 ,pp. 573 – 576, March 18-23, 2005

A Novel Supervised Dimensionality Reduction Algorithm for Online Image Recognition

Fengxi Song[1,2], David Zhang[3], Qinglong Chen[1], and Jingyu Yang[4]

[1] New Star Research Inst. of Applied Tech. in Hefei City, Hefei 230031, P.R. China
[2] Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, P.R. China
[3] Hong Kong Polytechnic University, Hong Kong, P.R. China
[4] Nanjing University of Science & Technology, Nanjing 210094, P.R. China
songfengxi@yahoo.com, csdzhang@comp.polyu.edu.hk,
ql_chen@sina.com, yangjy@public1.ptt.js.cn

Abstract. Image recognition on streaming data is one of the most challenging topics in Image and Video Technology and incremental dimensionality reduction algorithms play a key role in online image recognition. In this paper, we present a novel supervised dimensionality reduction algorithm—Incremental Weighted Karhunen-Loève expansion based on the Between-class scatter matrix (IWKLB) for image recognition on streaming data. In comparison with Incremental PCA, IWKLB is more effective in terms of recognition rate. In comparison with Incremental LDA, it is free of small sample size problems and can directly be applied to high-dimensional image spaces with high efficiency. Experimental results conducted on AR, one benchmark face image database, demonstrate that IWKLB is more effective than IPCA and ILDA.

Keywords: Dimensionality reduction, supervised learning, image recognition, streaming data, incremental algorithm.

1 Introduction

One of the most challenging problems in image recognition is the high dimensionality of an image space. The dimensionality of a high-resolution image is so large that conventional recognition algorithms are no longer technically feasible due to the curse of dimensionality and the heavy burden of computation. The result is that a high-dimensional image space has to first be compressed into a low-dimensional feature space, a procedure known as dimensionality reduction or feature extraction.

Dimensionality reduction algorithms are well studied in the past several decades. There are two main kinds of dimensionality reduction methods: unsupervised and supervised. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are typical examples of unsupervised and supervised dimensionality reduction methods respectively. In general, PCA is more suitable for concise representation or visualization of high-dimensional image data whereas LDA is more appropriate for pattern classification or recognition.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 198 – 207, 2006.

The typical implementation of various dimensionality reduction methods assumes that a complete training dataset is given in advance, and learning is carried out in one batch. However, in real-world applications of image recognition such as online image retrieval, robot vision, and surveillance, we often confront situations where a training set is not complete. Actually in most cases, data are presented as a stream of sample chunks. Streaming data are different from persistent data in that (a) they are transient, (b) usually they can only be read once, and (c) any systems working on them have no control over the order in which data streams arrive. To deal with streaming data, various incremental algorithms for unsupervised and supervised dimensionality reduction have been proposed.

Incremental Principal Component Analysis (IPCA) is a well studied technique and has a long history. Existing IPCA algorithms fall into two categories. The first category of IPCA [1-3] computes principal components directly from training samples by iteration. The second category of IPCA [4-6] computes principal components by performing matrix decomposition on an approximated total scatter matrix.

Due to the low effectiveness of IPCA algorithms, researchers pay more attentions to incremental supervised dimensionality reduction methods in recent years. Pang et al. proposed an Incremental Linear Discriminant Analysis (ILDA) algorithm for online face recognition [7]. Unfortunately, this ILDA method has some shortages. First, its updating scheme is memory-consuming. Second, it confronted with the so-called small sample size (SSS) problem and the strategy used to address SSS problem was not clearly stated. Later, Yan et al. proposed an Incremental Orthogonal Centroid (IOC) algorithm to extract discriminant features for text categorization [8]. IOC algorithm has two characteristics. First, it computes discriminant vectors directly from training samples by iteration. Second, its calculation procedure involves products of column vectors and row vectors both with high dimensionalities. The first characteristic of IOC leads to low effectiveness and poor efficiency for high-dimensional data, and the second property of IOC makes it inapplicable for high-resolution image recognition.

To overcome shortages of existing incremental feature extraction techniques, a new supervised dimensionality reduction algorithm for online image recognition—Incremental Weighted Karhunen-Loève expansion based on the Between-class scatter matrix (IWKLB) is proposed in this paper. In comparison with Incremental PCA, IWKLB is more effective in terms of recognition rate. In comparison with Incremental LDA, it is free of small sample size problems and can directly be applied to high-dimensional image spaces with high efficiency. Experimental results conducted on AR, one benchmark face image data-base, demonstrate that IWKLB is more effective than IPCA and ILDA.

2 Problem Definitions and Notations

For better comprehension, some important notations are introduced at first. Following that, the formal definition of our problem is given in this section.

2.1 Important Notations

Let $X = [\mathbf{x}_1,...,\mathbf{x}_N] \in R^{d\times N}$ be a data matrix of N training samples with c classes, where the ith training sample is represented as a d-dimensional column vector $\mathbf{x}_i$, and $\mathbf{n} = [N_1,...,N_c] \in R^c$ be the count vector whose elements are numbers of training samples from each class. It is obvious that

$$sum(\mathbf{n}) = \sum_{i=1}^{c} N_i = N\,. \tag{1}$$

Let I_i $(1 \le i \le c)$ denote the index set for samples from the ith class. Using the class average sample $\mathbf{m}_i = \frac{1}{N_i}\sum_{j\in I_i}\mathbf{x}_j$, the global average sample is given by

$$\mathbf{m} = \frac{1}{sum(\mathbf{n})} M\mathbf{n}^T = \frac{1}{N}\sum_{j=1}^{N}\mathbf{x}_j\,. \tag{2}$$

Here, M is the centroid matrix defined as

$$M = [\mathbf{m}_1,...,\mathbf{m}_c] \in R^{d\times c}\,. \tag{3}$$

The between- and the within-class scatter matrices are defined as follows:

$$S_b = \sum_{i=1}^{c} N_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T = H_b H_b^T \in R^{d\times d}\,. \tag{4}$$

$$S_w = \sum_{i=1}^{c}\sum_{j\in I_i}(\mathbf{x}_j - \mathbf{m}_i)(\mathbf{x}_j - \mathbf{m}_i)^T = H_w H_w^T \in R^{d\times d}\,. \tag{5}$$

Where

$$H_b = [\sqrt{N_1}(\mathbf{m}_1 - \mathbf{m}),...,\sqrt{N_c}(\mathbf{m}_c - \mathbf{m})] \in R^{d\times c}\,, \tag{6}$$

$$H_w = [\mathbf{x}_1 - \mathbf{m}_{k1},...,\mathbf{x}_N - \mathbf{m}_{kN}] \in R^{d\times N}\,,\ \mathbf{m}_{kj} = \mathbf{m}_i, \text{ if } j \in I_i\,. \tag{7}$$

2.2 Problem Definitions

Let us consider an initial training dataset $T_1 = \{\mathbf{x}_1,...,\mathbf{x}_N\} \subset R^d$ with class labels in the set of class label $L = \{1,...,c\}$, and streaming data $T_2 = \{\mathbf{x}(k),\ k = 1,2,...\} \subset R^d$. Formally, the online image recognition problem on streaming data consists of four steps: (1) Preprocessing, (2) Feature extraction, (3) Recognition, and (4) Updating. In the step of preprocessing, the initial training dataset is used to construct the initial discriminant model $\Omega = (H_w, H_b, M, \mathbf{n}, L)$. In the step of feature extraction, the discriminant matrix W is first derived from the current discriminant model and then used to compress the centroid matrix M and the new coming sample $\mathbf{x} \in T_2$ to $W^T M$ and $W^T \mathbf{x}$. In the step of recognition, $\arg\min_{1 \le i \le c} \left\| W^T (\mathbf{m}_i - \mathbf{x}) \right\|$ is predicted to be the class label of $\mathbf{x}$ based on the centroid or the minimum distance classifier. In the step of updating, the new coming sample $\mathbf{x}$ with its true class label $l(\mathbf{x})$ is joined in the training dataset, and the discriminant model $\Omega = (H_w, H_b, M, \mathbf{n}, L)$ is updated.

3 Incremental Weighted KLB

In this section, the concept of Weighted Karhunen-Loève expansion based on the Between-class scatter matrix (WKLB) is studied. Following that, an incremental algorithm of WKLB is presented.

3.1 Concept of WKLB

Karhunen-Loève (KL) expansion is widely used as a dimensionality reduction tool in data processing. Principal Component Analysis is actually a typical KL-expansion based on the total scatter matrix. Unlike PCA, which is an unsupervised feature extraction method, KL-expansion based on the Between-class scatter matrix (KLB) exploits the class label information. As a result, KLB is more effective than PCA in terms of recognition rate. In fact, as a feature extraction method, KLB is equivalent to Orthogonal Centroid (OC) [8] and its favorable performance has been confirmed by Park et al. [9]. The discriminant matrix of KLB, V consists of eigenvectors of S_b corresponding to nonzero eigenvalues.

KLB has two major advantages over other supervised feature extraction methods such as LDA for high-resolution image recognition. First, it is free of the SSS problem. That is, it can directly be applied to high-dimensional image space without the need to first apply other dimensionality reduction techniques such as PCA transformations in Fisherfaces [10] or pixel grouping in Null Space Method [11]. Second, by use of Singular Value Decomposition Theorem as in [12], its time- and memory-complexities are very low. Since KLB only uses the discriminant information between classes, however, its effectiveness might be promoted by exploiting the discriminant information within classes as well.

The Weighted KLB has two key points. First, the discriminant matrix of KLB, V is multiplied by an orthogonal matrix U. That is, discriminant vectors of KLB is rotated to eigenvectors of $\tilde{S}_w$. Second, each eigenvector of $\tilde{S}_w$ is weighted according to its corresponding eigenvalue.

The detailed calculation procedure of WKLB is as follows:

Step 1. Perform eigenvalue decomposition on the between-class scatter matrix S_b to obtain the discriminant matrix of KLB, V

Step 2. Map each sample vector $\mathbf{x}$ to obtain its intermediate representation $V^T\mathbf{x}$

Step 3. Perform eigenvalue decomposition on the within-class scatter matrix of project-ed samples, $\tilde{S}_w$ which is given by

$$\tilde{S}_w = V^T S_w V . \tag{8}$$

Let $\mathrm{N} = diag(\mu_1,...,\mu_r)$ be the eigenvalue matrix of $\tilde{S}_w$ in ascending order and $U = [\mathbf{u}_1,...,\mathbf{u}_r]$ be the corresponding eigenvector matrix. It follows that

$$U^T \tilde{S}_w U = \mathrm{N} . \tag{9}$$

Step 4. Choose a weighting function f and calculate the weighting matrix $f(\mathrm{N})$ using the following formulae

$$f(\mathrm{N}) = diag(f(\mu_1),...,f(\mu_r)) . \tag{10}$$

Step 5. Calculate the discriminant matrix of WKLB, W which is given by

$$W = VUf(\mathrm{N}) . \tag{11}$$

Since, the eigenvalue μ_i reflects the separability of samples when they are projected onto the projection vector $\mathbf{u}_i$. The smaller the eigenvalue μ_i, the better the projection vector $\mathbf{u}_i$. Thus, the weighting function f should be a non-increasing function and it should not overemphasize projection vectors with tiny eigenvalues and not over-depress projection vectors with huge eigenvalues.

In this paper, the following weighting function is used in all experiments.

$$f(\mu) = (1 + \log(1 + \mu))^{\alpha} . \tag{12}$$

Due to space limitation, detailed discussion on the selection of the weighting function is omitted from this paper.

3.2 IWKLB Algorithm

To describe the Incremental WKLB algorithm more clearly, we divide the algorithm into three sub-algorithms: preprocessing, feature extracting and updating as follows.

Algorithm 1.1. Preprocessing of IWKLB

```
Input: Data matrix $X = [\mathbf{x}_1, ..., \mathbf{x}_N]$ and class labels of the initial $N$ samples, $l(\mathbf{x}_1), ..., l(\mathbf{x}_N)$
    // $l(\mathbf{x}_j) \in L = \{1, ..., c\}$ is the class label of the jth sample $\mathbf{x}_j$
Output: Discriminant model $\Omega = (H_w, H_b, M, \mathbf{n}, L)$
1. Compute $H_b$, $H_w$, $M$, $\mathbf{n}$, and $L$ using the formula in section 2.1
```

Algorithm 1.2. Feature Extracting

```
Input: Precursors of the between- and within-class scatter matrix, $H_b$ and $H_w$
Output: Discriminant matrix of WKLB, $W$
1. Perform eigen decomposition to $H_b^T H_b$ as $H_b^T H_b = P^T \Lambda P$
2. Calculate the discriminant matrix of $S_b = H_b H_b^T$ using the formulae
       $V \leftarrow H_b P_b \Lambda_b^{-1/2}$.
   // Here $\Lambda_b$ is a diagonal matrix with all nonzero eigenvalues, $P_b$ the corresponding eigenvector matrix
3. Compute the within-class scatter matrix $\tilde{S}_w$ of projected samples using the formulae
       $\tilde{S}_w \leftarrow (V^T H_w)(H_w^T V)$
4. Perform eigen decomposition to $\tilde{S}_w$ as $\tilde{S}_w = U^T \mathrm{N} U$
5. Calculate the discriminant matrix of WKLB using formula (10-12)
```

Algorithm 1.3. Updating

```
Input: Discriminant model $\Omega = (H_w, H_b, M, \mathbf{n}, L)$, new training sample $\mathbf{x}$, and its class label $l(\mathbf{x})$
Output: Renewed discriminant model $\Omega = (H_w, H_b, M, \mathbf{n}, L)$
1. If $\mathbf{x}$ is from a newly introduced class, i.e. $l(\mathbf{x}) \notin L$
    2. Update the set of class label, $L$ using the formulae
           $L \leftarrow \{L, l(\mathbf{x})\}$
    3. Update the centroid matrix, $M$ using the formulae
           $M \leftarrow [M, \mathbf{x}]$
    4. Update the count vector, $\mathbf{n}$ using the formulae
           $\mathbf{n} \leftarrow [\mathbf{n}, 1]$
    5. Compute the global average sample $\mathbf{m}$ using the formulae (2)
    6. Update the precursor of the between-class scatter matrix, $H_b$ using the formulae
           $H_b \leftarrow [H_b, \mathbf{x} - \mathbf{m}]$
7. Else // Suppose $l(\mathbf{x}) = j$
    8. Update the precursor of the within-class scatter matrix $H_w$ using the formulae
           $H_w \leftarrow [H_w, \sqrt{\frac{N_j}{N_j + 1}}(\mathbf{x} - \mathbf{m}_j)]$
    9. Update the centroid matrix, $M$ using the formulae
           $\mathbf{m}_j \leftarrow \frac{N_j \cdot \mathbf{m}_j + \mathbf{x}}{N_j + 1}$
    10. Update the count vector, $\mathbf{n}$ using the formulae
           $N_j \leftarrow N_j + 1$
    11. Compute the global average sample $\mathbf{m}$ using the formulae (2)
    12. Compute the matrix, $H_b$ using the formulae (6)
13. End if
```

4 Performance Evaluation

To evaluate the performance of IWKLB, we compare the recognition rates of IWKLB, IPCA [6], and a refined version of Pang's ILDA [7] on the AR face image database when the Centroid classifier is used. In this section, we first discuss how to calculate the value of the parameter α in WKLB. Following that, we present experimental results of these three incremental dimensionality reduction algorithms.

4.1 Optimal Value of the Parameter Alpha

We try to experientially estimate the optimal value of α . In the following experiment, we use ORL face image database which contains 10 different images for 40 individuals. All images are grayscale and normalized with a resolution of 112×92. Five randomly selected images of each person are used for training and the remaining five for testing. Thus the total amount of training samples and testing samples are both 200. There is no overlapping between the training set and the testing set.

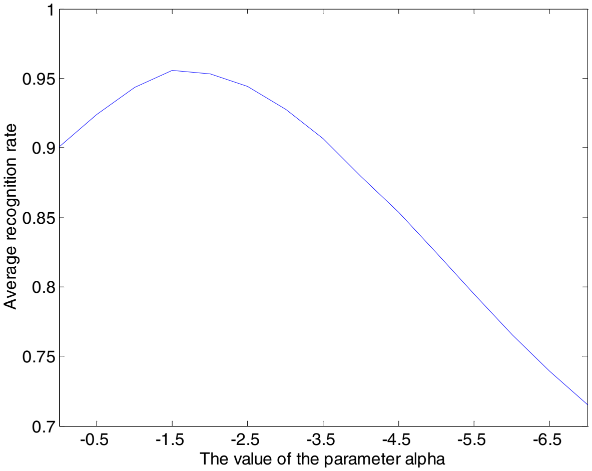


Fig. 1. Average recognition rate of WKLB vs. the value of the parameter α

Fig. 1 displays the curve of average recognition rate of WKLB with varying α over ten runs. Here, the centroid classifier with Euclidean distance is used in the experiment.

From Fig.1 we find that the optimal value of α is around -2. Apparently, the optimal value of α might be database-dependent. For simplicity, we let $\alpha = -2$ in the following experiments to evaluate IWKLB on AR face image database. An interesting fact is that although the parameter α has not been finely tuned, the recognition rates of IWKLB are significantly higher than those of IPCA, and ILDA as illustrated in Fig. 2.

4.2 Experimental Results

The subset of AR [13] face image database used in this paper contains 1680 face images of 120 individuals. All images are grayscale and normalized with a resolution of 50×40 and preprocessed using histogram equalization. In experiments, we randomly select one third of total samples, i.e. 560 (= 1680/3) samples as initial training samples and sequentially feed the remaining 1120 samples into IPCA, ILDA, and IWAS algorithms in a random order.

Fig. 2 displays trends of average recognition rates of various incremental facial feature extraction methods on the subset of AR of ten runs when the number of new training sample varies from 1 to 1050. Here the parameter α of IWKLB takes the value of -2.

We find that while the average recognition rates of ILDA and IWKLB increase with the number of new training samples, the average recognition rates of IPCA decline gradually. The probable reason is that the quality of the approximated total scatter matrix degenerates when the number of new training sample is increasing.

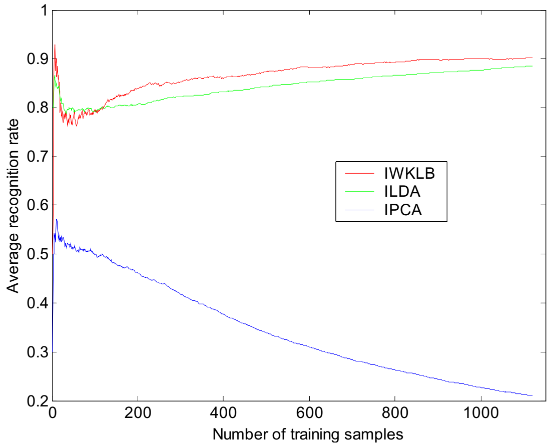


Fig. 2. Average recognition rates of IPCA, ILDA, and IWKLBAS vs. the number of new training sample on the AR face image database

5 Conclusions

We develop a new supervised dimensionality reduction algorithm—Incremental Weighted Karhunen-Loève expansion based on the Between-class scatter matrix (IWKLB) in this paper. In comparison with IPCA and ILDA, IWKLB is simple in theory and implementa-tion. Experiential studies demonstrate that IWKLB is a promising feature extraction algorithm for streaming data.

Acknowledgements

This work is supported by the National Science Foundation of China under Grant no. 60620160097, 60472060 and 60473039.

References

1. E. Oja and J. Karhunen, On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix, J. Math. Analysis and Application, vol. 106, pp. 69-84, Feb. 1985.
2. T.D. Sanger, Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network, IEEE Trans. Neural Networks, vol. 2, pp. 459-473, Nov. 1989.
3. Juyang Weng, Yilu Zhang, and Wey-Shiuan Hwang, Candid Covariance-Free Incremental Principal Component Analysis, IEEE Trans. Pattern Anal. Machine Intell., vol. 25, pp. 1034-1040, Aug. 2003.
4. P. Gill, G. Golub, W. Murray, and M. Saunders, Methods for modifying matrix factorizations, Mathematics of Computation, vol. 28, pp. 505–535, Apr.1974.
5. S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkeler, and H. Zhang, An eigenspace update algorithm for image analysis, Graphical Models Image Process, vol. 59, pp. 321-332, Sep. 1997.
6. Yongmin Li, On incremental and robust subspace learning, Pattern Recognition, vol. 37, pp. 1509-1518, Jul. 2004.

7. Shaoning Pang, Seiichi Ozawa, and Nikola Kasabov, Incremental linear discriminant analysis for classification of data streams, IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 35, pp. 905-914, Oct. 2005.
8. Jun Yan, Benyu Zhang, Ning Liu, et al., Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing, IEEE Trans. on Knowl. Data Eng., vol. 18, no. 3, Mar. 2006, pp.320-333.
9. H. Park, M. Jeon, and J. Rosen, Lower Dimensional Representation of Text Data Based on Centroids and Least Squares, BIT Numerical Math., vol. 43, pp. 427-448, 2003.
10. P.N. Belhumeur, J.P. Hespanha, and D.J. Kriengman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, IEEE Trans. on Pattern Anal. Machine Intell., vol. 19, no.7, pp. 711-720, Jul. 1997.
11. L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu, A New LDA-Based Face Recognition System Which Can Solve The Small Sample Size Problem, Pattern Recognition, vol. 33, no. 10, pp. 1713-1726, Oct. 2000.
12. Hua Yu and Jie Yang, A direct LDA algorithm for high-dimensional data—with application to face recognition, Pattern Recognition, vol. 34, no. 10, pp. 2067-2070, Oct. 2001.
13. A.M. Martinez and R. Benavente, The AR Face Database, CVC Technical Report, no. 24, Jun. 1998.

Feature Selection Based on Run Covering

Su Yang[1], Jianning Liang[1], Yuanyuan Wang[2], and Adam Winstanley[3]

[1] Shanghai Key Laboratory of Intelligent Information Processing, Dept. of Computer Science
and Engineering, Fudan University, Shanghai 200433, China
[2] Dept. of Electronic Engineering, Fudan University, Shanghai 200433, China
[3] National Centre for Geocomputation, Dept. of Computer Science, National University of
Ireland, Maynooth, Co. Kildare, Ireland
suyang@fudan.edu.cn

Abstract. This paper proposes a new feature selection algorithm. First, the data at every attribute are sorted. The continuously distributed data with the same class labels are grouped into runs. The runs whose length is greater than a given threshold are selected as "valid" runs, which enclose the instances separable from the other classes. Second, we count how many runs cover every instance and check how the covering number changes once eliminate a feature. Then, we delete the feature that has the least impact on the covering cases for all instances. We compare our method with ReliefF and a method based on mutual information. Evaluation was performed on 3 image databases. Experimental results show that the proposed method outperformed the other two.

1 Introduction

For pattern recognition problems, the data represented in feature space can be of very high dimensionality. However, some features are redundant and do not provide extra information over the others. In some worse cases, feature extraction could introduce noise, which does not contribute to pattern classification but degrade the classification performance. Thus, how to find a compact and effective feature subset is a significant issue, to which a great deal of effort has been devoted so far. There are two types of methodologies for dimensionality reduction: The unsupervised methods like PCA and the supervised methods, for which the class labels of the training samples are prior known. In this study, we foucse on the supervised dimensionality recduction, which is referred to as feature selection. Feature selection plays an important role in a variety of applications, including image classification [9,10]. Some reviews on feature selection can be found in [1-3]. According to [4], feature selectors can be sorted into two different groups: wrappers and filters. Wrappers employ a given classifier to evaluate features such that the feature selection is optimized for the given classifier. Filters evaluate features according to some measurements of class separability. In general, filters are less computationally complex than wrappers. As for filters, some methods measure the power of every independent feature in terms of class separability while some other methods measure the power of a subset of features as a whole. According to [3], only exhaustive search and the branch and bound methods

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 208 – 217, 2006.

[12,13] are optimal feature selectors. However, the branch and bound methods are based on an assumption that a performance index drops monotonously. In fact, investigations on developing new feature selectors have never stopped. Recently, mutual information based methods have received much attention [7,14-15].

In this study, we propose a new feature selection method, which belongs to the filter category. Its implementation is outlined as follows. First, the data at every attribute are sorted. The continuously distributed data with the same class labels are grouped into runs. The runs whose length is greater than a given threshold are selected as “valid” runs, which imply that the instances falling into such runs are separable from the other classes because enough instances from an identical class occupy spatially close positions. Second, we count how many runs cover every instance and check how the covering number changes once eliminate a feature. We delete the feature that has the least impact on the covering cases for all instances.

We compare our method with ReliefF [5], which is member of the Relief family [6], and the method based on mutual information [7]. Both methods belong to the filter category. We evaluate the 3 methods on 3 image databases provided in UCI Machine Learning Repository [16]. Experimental results show that the proposed method outperformed the other two.

2 The Method

The feature selection method is based on run covering. First, we sort the data values at every attribute. After the sorting, the data at every attribute can be divided into some segments, where the class labels of the elements in every segment should be identical. Such a segment is referred to as a run. If an instance is covered by at least one run (One of its attribute is included in the run.) whose length is greater than a given threshold, it means that this instance is separable from the other classes. By eliminating recursively such attributes that the removal of them will not affect the class separability in terms of run covering, a feature subset can then be selected. In the following, we first give the definition of runs. Then, we describe the feature selection algorithm. Finally, we provide a feature ranking method by which we can identify the least important feature and delete it in every loop.

2.1 Runs

The runs at every attribute can be extracted via the following procedure:

(1) Suppose that there are N instances. After sorting the kth attribute, we obtain $x_{k1}\le x_{k2}\le\ldots\le x_{kN}$. Denote the corresponding class labels as $[C(x_{k1}),C(x_{k2}),\ldots,C(x_{kN})]$. Note that $C(x_{ki})\in\{1,2,\ldots,L\}$, i=1,2,…,N, if there are L classes. Also, the indices of the corresponding instances are denoted as $[I(x_{k1}),I(x_{k2}),\ldots,I(x_{kN})]$.

(2) If $x_{ki}=x_{k,i+1}=\ldots=x_{k,i+U}$ but $C(x_{ki})=C(x_{k,i+1})=\ldots=C(x_{k,i+U})$ does not hold at the same time, it means that $x_{ki},x_{k,i+1},\ldots,x_{k,i+U}$ are not separable. To denote that, we let $C(x_{ki})=C(x_{k,i+1})=\ldots=C(x_{k,i+U})=0$. Note that only $0\notin\{1,2,\ldots,L\}$. Thus, it is not a valid class label.

(3) If $C(x_{ki})=C(x_{k,i+1})=\ldots=C(x_{k,i+U})\neq 0$, then, $[x_{ki},x_{k,i+1},\ldots,x_{k,i+U}]$ forms a run. The length of this run is $U+1$.

(4) Repeat (3) until all runs at every attribute have been found.

Some examples regarding the previously defined runs are shown in Fig. 1, 2, and 3, where the class labels distributed along a given attribute are illustrated. We can see that Fig. 1, 2, and 3 contains 2, 3, and 12 runs, respectively. Clearly, the case shown in Fig. 1 promises the best separability between the 2 classes while Fig. 3 corresponds with the worst case. The two cases shown in Fig. 1 and 2 are better in that the run length is greater. A longer run corresponds with a better case in terms of class separability. These examples show that the runs defined as above characterize the class separability to some extent. If the maximum run length at an attribute is too short as the case shown in Fig. 3, it means that the instances are not separable at this attribute. If we set a threshold of 5 and look for such runs whose length is greater this threshold, we can find out 2, 1, and 0 runs in Fig. 1, 2, and 3, respectively.

However, run length is a coarse characterization of class separability. It is known that N individually strong attributes are not certainly the best N attributes if combined together (N attributes performing well alone could perform unsatisfactorily as a team.). In this study, our focus is how to choose the best team, not the best N individuals. This can be achieved by using the run covering described in the next section.

1111111112222222222

Fig. 1. Class labels at a given attribute

22222111111111122222

Fig. 2. Class labels at a given attribute

1122112211221122112212

Fig. 3. Class labels at a given attribute

2.2 Eliminate Redundant Attributes Based on Run Covering

Prior to describing the feature selection algorithm, we give some definitions as follows.

(1) $R=\{R_i\}$: The run set including all the runs at every attribute.

(2) $\|R_i\|$: The length of the run $R_i\in R$.

(3) A: The attribute set that contains all remainder attributes following the feature elimination process described below. Initially, this set contains all the attributes. After the feature elimination process stops, the residual attributes are the finally selected features.

(4) /* Comments on pseudo codes */.

Following is the feature selection (feature elimination) algorithm:

(1) Assign a score to each attribute to represent the individual power of every attribute in terms of its contribution to class separability. Let us denote these scores as $w(1)$, $w(2)$, …., and $w(K)$. If $w(i)<w(j)$, it means that the ith attribute is better than the jth attribute in terms of class separability. This is also referred to as feature ranking. The detailed ranking algorithm is provided in section 2.3.

(2) Compute $C_l = \sum_{k,i} h_l(x_{kj}, R_i, T)$, where

$$h_l(x_{kj}, R_i, T) = \begin{cases} 1 & I(x_{kj}) = l \wedge x_{kj} \in R_i \wedge |R_i| > T \\ 0 & else \end{cases}.$$

/* If x_{kj} is a member of run R_i and the corresponding run length is greater than a threshold T, then, the corresponding instance $I(x_{kj})=l$ has been covered once. C_l corresponds with how many times the lth instance has been covered*/

(3) $\forall p \in A$, compute $C_{l,-p} = \sum_{k,i,k \neq p} h_l(x_{kj}, R_i, T)$.

/* The times that instance l has been covered without the kth attribute */

(4) Find $P = \{p : p \in A \wedge \sum_l |g(C_l) - g(C_{l,-p})| = 0\}$, where

$$g(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}.$$

/* P corresponds with the redundant attributes, the elimination of any of which will not cause a critical change on the times that each instance has been covered, where the critical change means that the covering times for any instance go down from a positive value to 0 suddenly after eliminating an attribute. */

(5) Find $q = \arg\max\{w(p) | p \in P\}$ and eliminate q from A.

/* Delete the least important feature in set P, where the criterion to select the least important feature refers to the feature ranking algorithm described in the next section */.

(6) If $P=\phi$, delete $q = \arg\max\{w(p) | p \in A\}$

/* If no feature satisfying that elimination of it will not change the covering case for every instance, then, delete the least important feature ranked by the feature ranking algorithm described in the next section. */

(7) Let $C_l=C_{l,-p}$ and Go to (3) until the number of the residual attributes in A is equal to the predefined desired number.

Some discussions about the above algorithm are given below. The central idea of this algorithm is: Look for such attributes that the class separability will not be affected if eliminating them. The run covering plays an important role in this algorithm. First, we select the runs whose length is greater than a given threshold T. Every selected run covers the instances that are separable at a given attribute since the instances from the same class distribute very closely to each other (They are within a run). As every instance has K attributes, it has the chance to be covered by K runs at most. If an instance is covered $V \leq K$ times by the runs, then, eliminating one attribute from the V attributes will not affect the classification of this instance because it is still covered by the runs at the other V-1 attributes, which means that this instance is still close to the instances from the same class at the V-1 attributes. Taking all the instances into account together, we hold the following idea. Suppose that $Q \leq N$ instances are covered by at least one run. When we eliminate one attribute, if the Q covered instances are still covered by at least one run, then, it means that this attribute is redundant and contributes no additional information in contrast to the reminder attributes. Eliminating it should have no impact on the classification. In case there exist R>1 attributes that the removal of any of them will not chance the covering, we eliminate only one attribute among the R attributes and then recompute the covering. In such a case, the selection of the attribute to be eliminated is not random. It is based on a feature-ranking criterion. That is, we firstly score every attribute according to its individual significance in terms of class separability. Then, we always eliminate the least important one from the R attributes. The feature-ranking criterion is described in detail in the next section. The above procedure can be repeated to eliminate redundant attributes recursively.

In the above algorithm, T is the only parameter (See step 2), which determines how many runs are valid in counting the covering number. We let the threshold $T=0.1\times N$, where N denotes the number of all instances. We have tested a couple of different values for T and found that $T=0.1\times N$ is a satisfactory one in this study, which not only leads to a satisfactory overall classification performance but also promises a stable classification performance when $T\in[0.1\times N\text{-}\Delta,0.1\times N+\Delta]$, where Δ is a relative small positive value. Note that T can be scaled to adapt to problmes from different domains.

The above algorithm can be easily extended to multi-class classification. We only need to decompose the multi-classification into multiple two-class classifications (pairwise classification). Then, we look for such attributes the elimination of which do not affect the covering for every two-class classification. For example, if there are L classes, then, we decompose the L-class separability computation into $L(L\text{-}1)/2$ parallel two-class separability computations. Here, step (1)~(3) and step (7) are implemented as $L(L\text{-}1)/2$ parallel processes. In step (4), the intersection of the $L(L\text{-}1)/2$ solutions forms P. The other steps are the same as described prevoiusly.

$\times\times\times\times\times\times\times\underline{+}\,\underline{\times}++++++$

Fig. 4. Distribution of two classes along a given attribute

$\times\times\times\times\times\times\underline{+}\,\underline{+}\,\underline{\times}\,\underline{\times}+++++$

Fig. 5. Distribution of two classes along a given attribute

2.3 Feature Ranking

Suppose that there are M and N samples in class X and Y and the kth attribute of the two classes are $\{x_{k1},x_{k2},\ldots,x_{kM}\}$ and $\{y_{k1},y_{k2},\ldots,y_{kN}\}$, respectively.

We define the relationship between x_{ki} and y_{kj} as

$$H(x_{ki},y_{kj})=\begin{cases}1 & x_{ki}<y_{kj}\\ 0 & x_{ki}\geq y_{kj}\end{cases}. \tag{1}$$

The above definition means that if x_{ki} lies in the left side of y_{kj}, then, $H(x_{ki},y_{kj})=1$. Else, $H(x_{ki},y_{kj})=0$.

Based on the relationship between two instances, we define the overall relationship between the two classes in terms of the kth attribute as

$$d_k=\max\{\sum_{i=1}^{M}\sum_{j=1}^{N}H(x_{ki},y_{kj}),\sum_{i=1}^{M}\sum_{j=1}^{N}H(y_{kj},x_{ki})\}. \tag{2}$$

It summarizes the relationship between every class X sample and every class Y sample. Also, it reveals the separability between the two classes and can be understood as a distance measure between the two classes. This is explained via the following two examples.

See the example shown in Fig. 4, where the samples in the overlapping region are underlined. Suppose that, in the from left to right order, the "×" marks represent one-dimensional class X samples $x_1,x_2,\ldots,x_8$ and the "+" marks represent one-dimensional class Y samples $y_1,y_2,\ldots,y_7$, respectively. The underlined "×" corresponds with x_8 and the underlined "+" corresponds with y_1. With regard to x_1, all the 7 samples of the other class lie in the right side of it. So, we obtain $\sum_j H(x_1,y_j)$=7. With regard to x_8, only 6 samples of the other class lie in the right side of it. Thus, we hold $\sum_j H(x_8,y_j)$=6. In fact, $\sum_j H(x_i,y_j)$ figures out how many samples in class Y locate in the right side of x_i. In contrast, $\sum_j H(y_j,x_i)$ reveals how many samples in class Y locate in the left side of x_i. Therefore, $\sum_i\sum_j H(x_i,y_j)$ is a measure of the degree that class X locates in the left side of class Y and $\sum_i\sum_j H(y_j,x_i)$ characterizes the degree that class X locates in the right side of class Y. Obviously, $\max\{\sum_i\sum_j H(x_i,y_j), \sum_i\sum_j H(y_j,x_i)\}$ reveals the relative relationship between the two classes of interest. For the above example, $\sum_i\sum_j H(x_i,y_j)$=55 and $\sum_i\sum_j H(y_j,x_i)$=1. This means that most samples of class X locate in the left side of class Y. In accordance with Eq. (1), the separability measure between the two classes is 55. Now, consider another example shown in Fig. 5, where the overlapping region is larger than the case shown in Fig. 4. Correspondingly, the separability measure between the two class computed via Eq. (1) is 52. Taking into account the two examples, it is easy to see that a smaller separability measure corresponds with a more severe overlap between the two classes of interest, namely, a worse case in terms of separability. On the contrary, a greater separability measure, which corresponds with a smaller overlapping degree, means a better case in terms of separability.

Suppose that there are L classes and class j contains $N(j)$ samples, j=1,2,…,L. Let $x_{ki}^{(j)}$ denote the kth attribute of the ith sample of class j. We further assume that every sample has K attributes. The feature-ranking algorithm is described below. Suppose that the input is $\{ x_{ki}^{(j)} \mid j=1,2,\ldots,L;\ i=1,2,\ldots,N(j);\ k=1,2,\ldots,K\}$. With regard to the kth attribute, compute the separability between every pair of classes via Eq. (1) and Eq. (2), that is, $\{ d_k^{(u,v)} \mid u=1,\ldots,L\text{-}1;\ v=u+1,\ldots,L\}$. Then, let $\sum_{u,v} d_k^{(u,v)}$ be the overall discrimination power of the kth attribute, according to which all attributes can be ranked.

2.4 Computational Complexity

Suppose every class contains N samples. Let L denote the class number, K the feature number, and M the dimension of set A. The complexity of step 1, step 2, and the loop from step 3 to step 7 is roughly $O(K\times L\times(L\text{-}1)\times N^2)$, $O(L\times(L+1)\times K\times N)$, and $O(M\times(M+1)\times L\times(L+1)\times N)$, respectively. The overall complexity is basically the sum of the three parts.

3 Experimental Results

We tested the proposed algorithm with UCI machine learning databases [16]. The performance evaluation was conducted with the letter recognition database, the satellite image classification database, and the image segmentation database. The data properties of the 3 databases are summarized in Table 1. We also compare our method with 2 other methods: ReliefF [5] and the method based on mutual information [7]. In classifying every data set, we use 3 classifiers: 1-nearest neighbor (1-NN), decision tree, and support vector machine (SVM). Here, we use the weka software to implement Relief and the decision tree as well as the SVM classifier [17]. We apply 10-fold cross validation for performance evaluation [8].

The classification accuracy against the feature number for the image segmentation data is illustrated in Fig 6, 7, and 8, where 1-NN, decision-tree, and SVM classifiers are applied, respectively. Obviously, the proposed method outperforms the other two methods. For the 1-NN classification based on the proposed feature selector, when the feature number is equal to 3, the classification accuracy reaches 97.23%. Then, the classification accuracy changes very little, between 96.49% and 97.58%. The classification accuracy using the full attributes is 96.62%, which is less than that using only 3 features selected by the proposed algorithm. See Fig. 6, the other two methods perform much worse than the proposed method. See Fig. 7 and Fig. 8, the same case takes place when comparing the 3 methods based on decision tree and SVM classification.

The classification accuracy against the feature number for the satellite image data is shown in Fig 9, 10, and 11, where 1-NN, decision-tree, and SVM classifiers are applied, respectively. It can be seen that the proposed method outperforms the other two methods given any feature number.

The classification accuracy against the feature number for the letter recognition data is exhibited in Fig 12, 13, and 14, where 1-NN, decision-tree, and SVM classifiers are applied, respectively. The proposed method promises comparable performance to ReliefF while both methods outperform the method based on mutual information.

In the above 3 benchmarks, we can see that different classifier leads to different classification performance but the comparison among different feature selection methods never changes with the choice of classifiers. According to Fig. 9~11, the proposed method approaches the best performance or a satisfactory perofrmance very quickly but the other two methods do not. The above comparisons show that the proposed method performs well in selecting useful features for image classification.

Table 1. Data properties

Data	#Attributes	#Instances	#Classes
Image	19	2310	7
SatImage	36	6435	6
Letter	16	20000	26

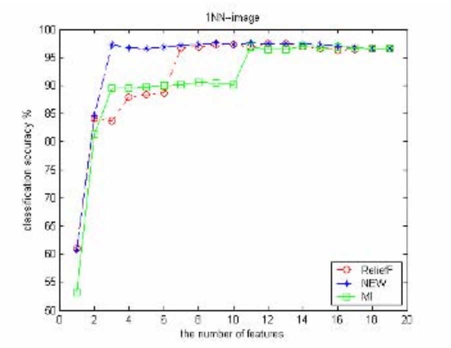


Fig. 6. Classification accuracy against feature number using 1-NN: image segmentation

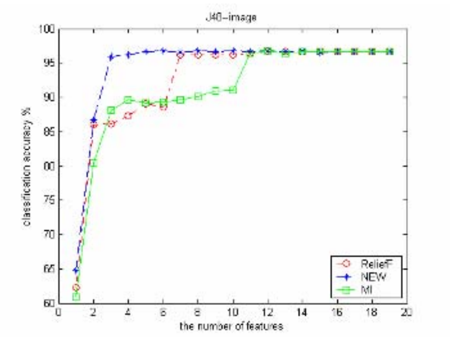


Fig. 7. Classification accuracy against feature number using decision tree: Image segmentation

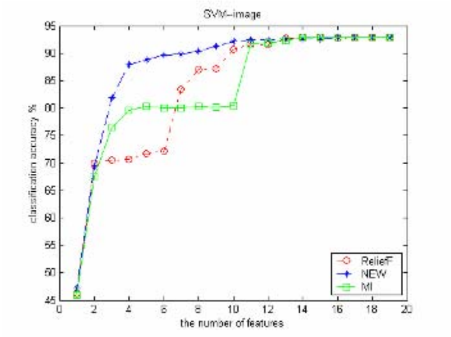


Fig. 8. Classification accuracy against feature number using SVM: Image segmentation

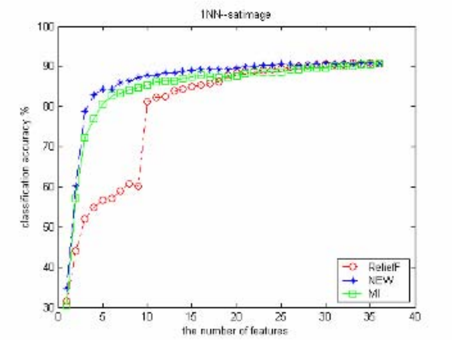


Fig. 9. Classification accuracy against feature number using 1-NN: Satellite image

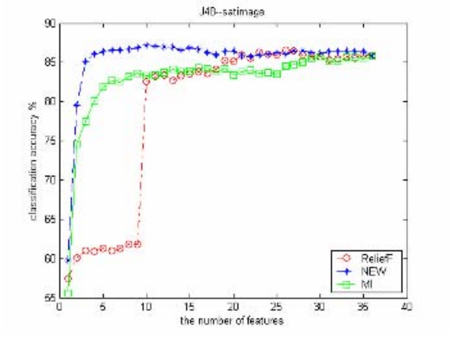


Fig. 10. Classification accuracy against feature number using decision tree: Satellite image

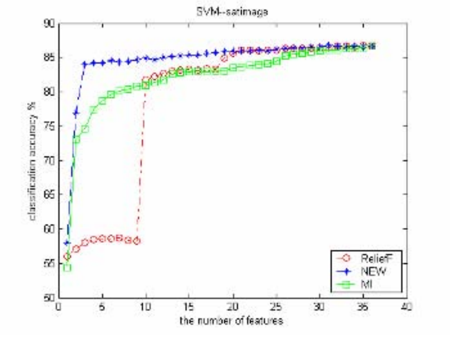


Fig. 11. Classification accuracy against feature number using SVM: Satellite image

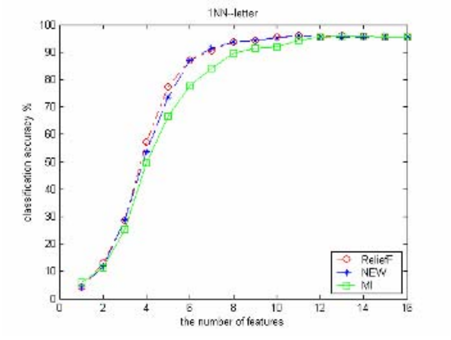


Fig. 12. Classification accuracy against feature number using 1-NN: Letter

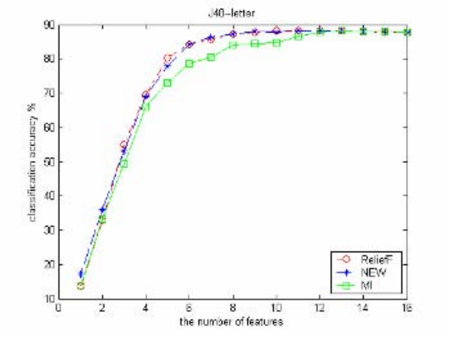


Fig. 13. Classification accuracy against feature number using decision tree: Letter

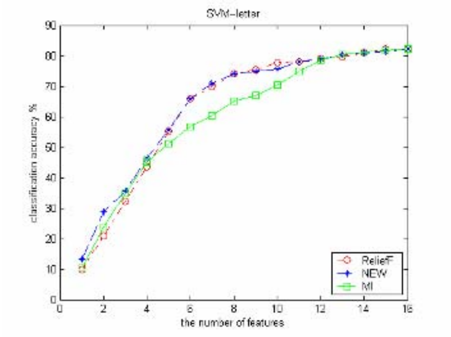


Fig. 14. Classification accuracy against feature number using SVM: Letter

4 Concluding Remarks

In this study, we propose a new feature selection method. It is based on run covering. The heart of this algorithm is to check whether the removal of a given attribute will change the covering of every instance. If not, it can be decided that this attribute is redundant. The run length plays an important role in judging whether an instance is separable from the other classes at a given attribute. The experiments confirmed the effectiveness of this method in terms of selecting useful features for image classification. Note that the run-length based method works with not only the linear separable attributes but also the attributes that are not linearly separable.

Another important issue is the stopping criterion, that is, what feature number is satisfactory to stop the feature elimination procedure. For the limited space of this paper, we did not present the criterion and the related performance evaluation. One stopping criterion can be: If the covering case for any instance changes after eliminating a feature, then, stop the feature selection. It is easy to implement. We just need to modify step (6) of the algorithm to be: If $P=\phi$, the desired feature number has been approached.

Acknowledgement. This work is supported in part by Natural Science Foundation of China under grant 60305002, China/Ireland Science and Technology Research Collaboration Fund under grant CI-2004-09, and National Basic Research Program of China under grant 2006CB705700.

References

[1] Guyon, I. and Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3 (2003) 1157-1182

[2] Liu, H. and Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowledge and Data Engineering, 17 (2005) 491-502

[3] Jain, A. and Zongker, D.: Feature selection: Evaluation, application, and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19 (1997) 153-158

[4] Kohavi, R. and John, G. H.: Wrappers for feature subset selection. Artificial Intelligence, 97 (1997) 273-324

[5] Robnik-Sikonja, M. and Kononenko, I.: Theoretical and empirical analysis of ReliefF and RreliefF. Machine Learning, 53 (2003) 23-69

[6] Kira, K. and Rendell, L.: A practical approach to feature selection. Proc. Int. Conf. Machine Learning, (1992) 249-256

[7] Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (2005) 1226-1238

[8] Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. Proc. Int. Joint Conf. Artificial Intelligence, (1995) 1137-1145

[9] Y. Rui, T. S. Huang, and S. Chang, "Image retrieval: Current techniques, promising directions and open issues," Visual communication and image representation, vol. 10, no. 4, pp. 39-62, 1999.

[10] Ho, T. K and Baird, H. S.: Pattern classification with compact distribution maps. Computer Vision and Image Understanding, 70 (1998) 101-110

[11] Cover, T. M.: The best two independent measurements are not the two best. IEEE Transactions on Systems, Man, and Cybernetics, 4 (1974) 116-117

[12] Narendra, P. M. and Fukunaga, K.: A branch and bound algorithm for feature subset selection. IEEE Transactions on Computers, 26 (1977) 917-922

[13] Somol, P., Pudil, P., Kittler, J.: Fast branch & bound algorithms for optimal feature selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), 900-912

[14] N. Kwak and C. H. Choi, "Input feature selection by mutual information based on Parzen windows," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 12, pp. 1667-1671, 2002.

[15] Trappenberg, T., Ouyang, J., Back, A.: Input variable selection: Mutual information and linear mixing measures. IEEE Trans. Knowledge and Data Engineering, 18 (2006) 37-46

[16] http://www.ics.uci.edu/~mlearn/MLRepository.html

[17] http://www.cs.waikato.ac.nz/~ml

Automated Detection of the Left Ventricle from 4D MR Images: Validation Using Large Clinical Datasets

Xiang Lin[1], Brett Cowan[2], and Alistair Young[1,2]

[1] Bioengineering Institute, University of Auckland, New Zealand
[2] Center for Advanced MRI, University of Auckland, New Zealand
{x.lin, b.cowan, a.young}@auckland.ac.nz

Abstract. We present a fully automated method to estimate the location and orientation of the left ventricle (LV) from four-dimensional (4D) cardiac magnetic resonance (CMR) images without requiring user input. The method is based on low-level image processing techniques which incorporate anatomical knowledge and is able to provide rapid, robust feedback for automated scan planning or further processing. The method relies on a novel combination of temporal Fourier analysis of image cines and simple contour detection to achieve a fast localization of the heart. Quantitative validation was performed using two 4D CMR datasets containing 395 patients (63720 images), with a range of cardiac and vascular disease, by comparing manual location with the automatic results. The method failed in only one case, and showed an average bias of better than 5mm in the apical, mid-ventricular and basal slices in the remaining 394. The errors in the automatically detected LV orientation were similar to those found in scan planning when performed by experienced technicians.

1 Introduction

We investigated methods for the robust, accurate and fully automatic identification of heart location and orientation from CMR examinations. The method is targeted at clinical applications and must therefore be fast, efficient and reliable. It should be able to return the location, orientation and approximate contours of the LV in the absence of any user input. The method is expected to have two important applications. Firstly, the detected LV contours could be used as input to higher level segmentation methods including deformable model based analyses. Secondly, the method could be used to speed up image acquisition by facilitating the fully automatic planning of CMR examinations.

Segmentation of the LV in CMR images is important for quantitative assessment of cardiac function and many automatic approaches at different levels of image processing have been proposed to tackle this problem. Low level techniques depending on local image intensity characteristics are fast but lack robustness. *A priori* knowledge can be incorporated into deformable model-based approaches; however, the efficiency and robustness of these methods is heavily dependent on

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 218–227, 2006.

the initial contours or models. Most semi-automatic methods such as [1] require manual initialization. Fully automatic algorithms have been proposed but many of these are computationally intensive [2] or lack a wide range of clinical validation [3]. One [4] has been validated in 121 cases but assumes that the location of the heart is approximately at the center of the MR image. Specialized methods have also been proposed for tagged [5] and perfusion images [6], however their application to patients with a wide range of clinical disease remains uncertain.

Automated CMR image planning has been proposed as a strategy for speeding up scan acquisition [7,8]. The core requirement is for a fast and accurate calculation of the three-dimensional (3D) position and orientation of the LV. A deformable template based method [7], which estimated the LV axis by fitting many feature points of major thoracic organs in the localizer images, was computationally intensive. To avoid this problem, [8] proposed another method which employed *a priori* knowledge of the average LV direction to speed up the procedure. The scout images were then segmented by thresholding and both the LV and right ventricle (RV) were localized by comparison with morphologic characteristics of the candidate objects. However, in our experience clinical image variability compromises the robustness of this method.

We relied on simple methods to automatically estimate heart location and orientation in order to provide rapid feedback to higher level processes. The assumptions of our method are listed below. Any cases which violate these assumptions (eg congenital heart disease in which the LV and RV are transposed) would not be expected to be solved by our method.

1) The heart is the only large organ in the thorax with a temporal fundamental frequency equivalent to the cardiac cycle.
2) The orientation of the heart is similar across a wide variety of (non-congenital) cardiac diseases (this assumption is validated below).
3) The short axis (SA) slices are ECG gated and have been planned approximately orthogonal to the long axis of the LV (we show that this assumption is not restrictive in practice).
4) The positions of the LV in adjacent slices are spatially and temporally coherent.
5) The septal myocardium (heart muscle) is close to the centroid of the heart and has the LV and RV blood pools on each side. The boundary between the LV blood pool and septal myocardium is not degraded by large papillary muscles or trabeculations (which are not typically expected in this area anatomically).

The reminder of the paper is organized as follows. In Sect. 2, we describe the details of our method. In Sect. 3, we present the results from 395 patients from two independent clinical trial datasets. The conclusion is provided in Sect. 4.

2 Method

Our method is based on the novel combination of the Fourier transform (FT) in the temporal domain with *a priori* orientation and shape information in space.

The FT is employed to calculate an average (DC) image and first harmonic (H1) magnitude image for each cine slice. Even in severely diseased hearts, this method successfully identifies the heart in most cases. The output of the FT is then used to derive a region of interest (ROI) and the threshold level which robustly delineates the LV. This four step process is summarized below:

1) Organize the frames for each slice and apply the FT over time to obtain the DC and H1 images for each slice (Sect. 2.2).
2) Compute a ROI for each slice and the centroid for the whole heart from the H1 images (Sect. 2.3).
3) Find a pixel on the septal myocardium and compute the threshold level to delineate blood from myocardium in the DC images (Sect. 2.4).
4) Threshold the DC images and locate the LV on all slices (Sect. 2.5).

2.1 Patient Data and Ground Truth

Two clinical datasets are utilized in this study. The ONTARGET (Ongoing Telmisartan Alone and in combination with Ramipril Global Endpoint Trial) dataset contained 330 patients with cardiac and vascular disease recruited from 10 MR centers world-wide as enrolled in the CMR substudy to ONTARGET [9]. This study was the source of the *a priori* heart orientation information which is integrated into our method. Data from the second trial known as ZEST (New Zealand Eplerenone aortic Stenosis Trial) was used for independent validation purposes.

ONTARGET Dataset. The 330 patients had a range of disease histories: 294 had coronary artery disease, 46 had peripheral arterial disease, 111 had diabetes, 202 had hypertension and 192 had suffered a previous myocardial infarction (with the total exceeding the number of patients due to multiple diagnoses). The patients were recruited in six countries and imaged using standard SSFP cardiac cine sequences on Siemens, Philips and GE scanners. Either prospectively or retrospectively gated images were acquired in six equally spaced SA locations from apex to base. Typical imaging parameters were TR / TE / flip / FOV = 30 ms / 1.6 ms / 60° / 360 mm, slice thickness 6 mm, image matrix 256×208. There were typically 25 temporal frames per slice, depending on the heart rate. All cines were acquired during breath-holding of 8–15 seconds duration.

ZEST Dataset. The ZEST dataset was collected for the purpose of determining the treatment effect of Eplerenone in patients with asymptomatic moderate or severe aortic stenosis (to be published). 65 patients were scanned at nine centers within New Zealand for the primary assessment of LV mass. The image data used for validation in this study was collected during the baseline visit. The imaging parameters were similar to the ONTARGET trial.

Ground Truth. The ground truth for the heart location and orientation was determined manually by two experienced technicians operating independently

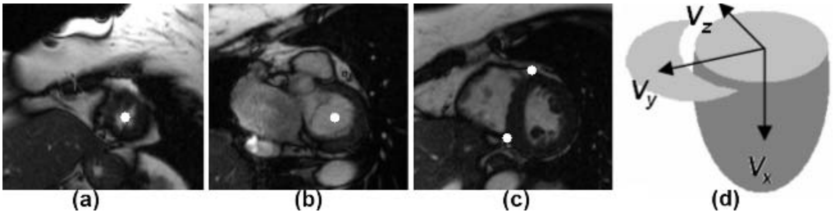


Fig. 1. Manual definition of heart orientation (a) center of the LV on an apical SA slice, (b) center of the LV on a basal SA slice, (c) RV insertion points for defining V_y and (d) right handed coordinate system

on the end-diastolic images. The 3D orientation of the LV long axis (V_x) was defined by two points manually placed in the middle of the LV blood pool at the apex and the base respectively (Fig. 1a and 1b). The orientation of the RV (V_y) was defined by the centroid of points placed on the endocardial insertion of the RV on all SA slices showing the RV (Fig. 1c). The remaining axis (V_z) was oriented posteriorly to complete a right handed coordinate system (Fig. 1d) [10]. The average directions $\overline{V}_x$, $\overline{V}_y$ and $\overline{V}_z$ from all ONTARGET cases were then computed for use in the automated method below.

2.2 Fourier Transform over Time

The heart is the only large structure in the thorax with substantial motion at a frequency given by the heart rate, and this characteristic makes the heart distinguishable by analyzing changes in pixel intensity. Figure 2b shows two typical pixel intensities through time. P_{in} is a pixel at the boundary between the LV blood pool and the septal myocardium and its intensity changes through a large range over time. P_{out} is also located close to the boundary of two different structures but is relatively static. Previously the standard deviation of pixel intensity has been used to locate the heart [11,12], however we found that in around 20% of cases the standard deviation images were contaminated by excessive high frequency noise (Fig. 2g). The differences between P_{in} and P_{out} are most clearly appreciated in the magnitude of the first harmonic (H1) component of the FT (Fig. 2c), even though their DC components are very similar (Fig. 2d). We therefore computed the FT for every pixel in the image and then used the DC component (Fig. 2e) and H1 image (Fig. 2f) in the subsequent analysis. This method provides excellent delineation of the cardiac structures, as well as the great vessels such as the aorta.

2.3 Fast ROI Analysis

A cardiac centroid and region of interest (ROI) containing the heart were calculated from the H1 images for each slice as follows. Firstly, in order to reduce the effect of noise and signal from non-cardiac structures, the H1 images were

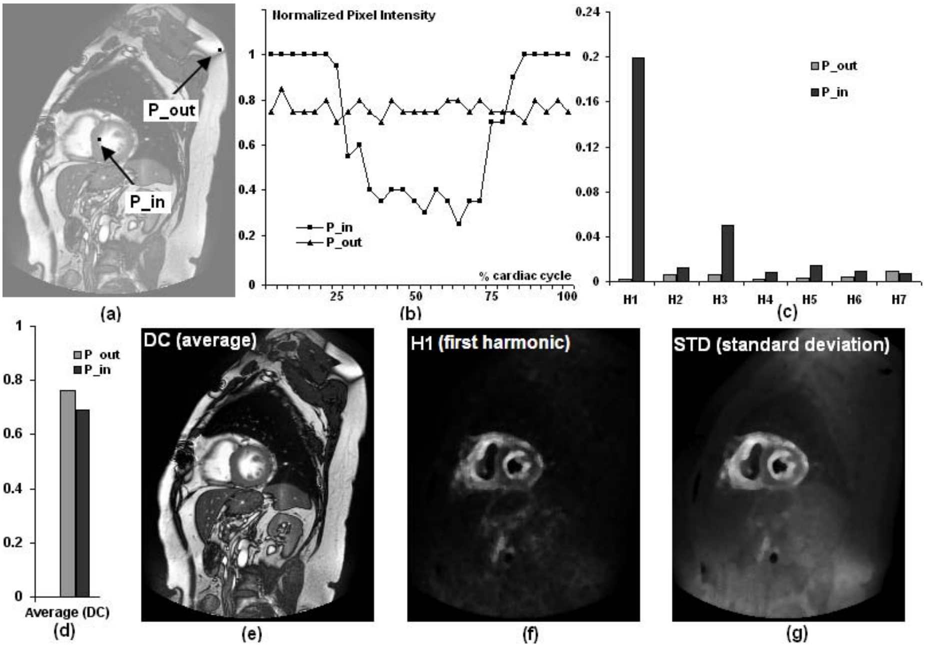


Fig. 2. Temporal Fourier transforms for each pixel in the time sequence (a) image showing a pixel near a moving boarder inside the heart (P_{in}) and a pixel near a stationary boarder (P_{out}), (b) pixel intensity versus time, (c) comparison of the magnitude of the first seven frequency components for P_{in} and P_{out}, (d) comparison of the DC components, (e) DC (average) image, (f) H1(first harmonic) image and (g) standard deviation image

filtered with a smoothing filter and all pixels with a magnitude less than 5% of the maximum magnitude within the 3D volume were set to zero. Secondly, the ROI for each slice was iteratively refined. For each iteration, the centroid of the H1 image was computed for each SA slice. A 3D line was then fitted to the centroids of all SA slices by linear least squares. A distance distribution of all H1 pixels to the 3D line was calculated and weighted with each pixel's intensity value. A Gaussian curve was then fitted to this distribution and all pixels greater than a certain distance from the line were removed. The cut position to define this cylinder of interest y was calculated using Eq. 1:

$$y = \mu + \sqrt{2} erf^{-1}(x)\sigma \tag{1}$$

where x is the percentage of the pixels the cylinder should include (95% on our experiments), μ is the mean and σ is the standard deviation of the Gaussian distribution. The 3D centroid of the H1 volume was then computed and compared to the previous 3D centroid. Iteration terminated when the distance between successive 3D centroids was less than one pixel. In most cases, the

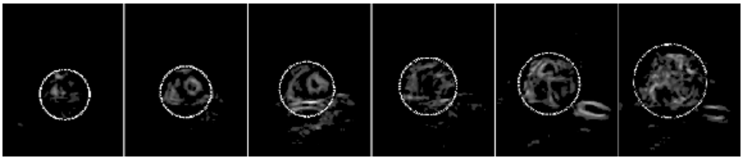


Fig. 3. Result of the ROI determination for each slice (apex to base from left to right)

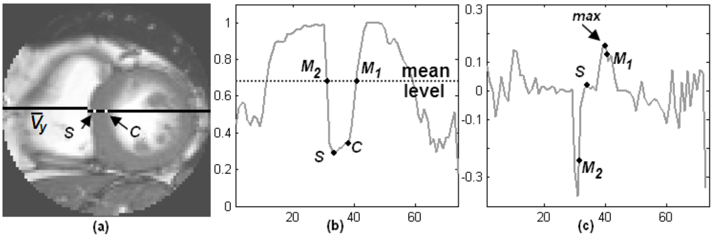


Fig. 4. Calculation of the threshold level (a) search line for threshold shown on the ROI image, (b) intensity for each pixel showing local minimum for the septum S, (c) intensity gradient for each pixel showing the position of the maximum gradient (max)

iteration terminated after only one loop. Finally, the ROI was adjusted on each slice individually using the Gaussian fitting method to produce circular regions of interest of appropriate diameter on each slice. The results are shown in Fig. 3.

2.4 Parameters for Blood Pool Segmentation

In order to provide an initial segmentation of the LV blood pool, as well as a separation of the RV and LV blood pools, we used the DC images cropped by their respective circular ROI (Fig. 4a) to locate a pixel within the septal myocardium. The threshold level which best discriminated the blood and myocardial signals was then calculated as follows. Firstly, the mid-ventricular SA slice closest to the 3D centroid was chosen. The center of the ROI was obtained by intersecting the 3D least squares line (from Sect. 2.3) with the slice, marked C in Fig. 4a. This point is almost always close to the interventricular septum. A line passing through C was defined in the average direction of the RV ($\overline{V}_y$). The intensity of the DC image along this line (Fig. 4b) was then used to locate the septum by searching for a local minimum within the region where the curve was less than the average intensity level (M_1 and M_2 are the two intersection points between the average intensity level and the curve in the neighbourhood of S). Once a septal point S was found, the LV could be located on the $-\overline{V}_y$ side. The blood pool threshold level was then determined by searching for the pixel with the maximum gradient between M_1 and M_2 (max in Fig. 4c). To avoid the noise and uncertainty inherent in analyzing only a single line, we also analyzed eight additional lines parallel to $\overline{V}_y$ and computed the average value of these results.

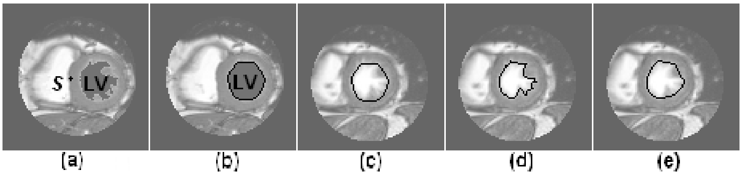


Fig. 5. Locating the LV blood pool (a) LV blood pool detected on the middle slice by thresholding, (b) convex hull applied to the middle slice, (c) projection of the LV blood pool onto an adjacent slice, (d) thresholding and selection of the most similar binary object as the detected LV blood pool, and (e) convex hull applied to the new slice

2.5 LV Detection

The LV blood pool in the middle slice (defined as the slice closest to the 3D centroid from the H1 volume) was localized by thresholding on the $-\overline{V}_y$ side of S, as shown in Fig.5a. A convex hull (Fig. 5b) was then used to reduce the impact of the papillary muscles, as in many other papers (e.g. [5,8]).

To find the LV blood pool in adjacent slices, we modified the method proposed in [5]. The analysis was based on binary images created by thresholding, and assumed that the LV regions are spatially coherent between slices. The LV blood pool detected in the middle slice was projected to its two neighboring slices and the binary objects obtained by thresholding on the neighboring slices compared with it. Rather than project the region in the direction normal to the slice [5], we projected in the average long axis direction $\overline{V}_x$ in order to improve robustness to the orientation of the image planes. The binary object most similar to the projection in each slice was then selected. The similarity of the two objects was calculated by the area of the intersection divided by the area of union [5]. Figure 5b is the middle slice with the detected LV blood pool superimposed on it. The region is projected to its neighboring slice (Fig. 5c) and the most similar object is then found (Fig. 5d). Finally, the convex hull is applied to the new region (Fig. 5e).

With this method, the LV regions on all slices were located (Fig. 6). The similarity between the projected and binary regions could be very low on the basal slice because of the leakage of the blood pool region during thresholding. In such cases, an erosion operation was iteratively used to improve the leakage. If the operation could not satisfy the requirement of a maximum number of

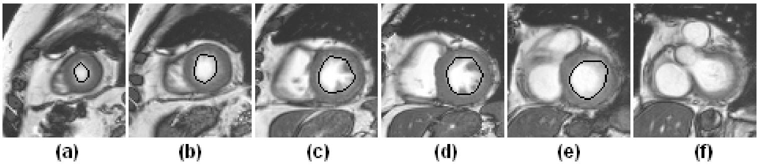


Fig. 6. Example of the detected LV blood pool on all SA slices (apex to base from left to right)

iterations, then no region was reported, as is shown in Fig. 6f. A 3D line was then fitted to the centroids of the resulting LV regions to define the final V_x.

3 Results

The fully automated method was implemented in Matlab and required approximately 4.13 seconds (not compiled) to run on a PC (Pentium IV 3.2GHz) for each case, excluding the DICOM file reading time. The first experiment was performed on the 330 cases in the ONTARGET dataset which had initially been used to define the average $\overline{V}_x$, $\overline{V}_y$ and $\overline{V}_z$ directions. The algorithm failed to detect the LV in only one case, where it found the RV. To validate the robustness of the method, it was then tested against the ZEST dataset. It contained 65 independent cases which had not been used in any way during the development of the method. There were no failures in this group.

Errors between manual and automatic methods are reported below. The ONTARGET evaluation included only the 329 successful cases.

(a) Angular Errors. We first investigated the inter-observer error in ground truth by determining the average difference in V_x between Observer A and Observer B, which was 3.5 ± 2.4 degrees. We also computed the difference between the mean directions $\overline{V}_x, \overline{V}_y$ and $\overline{V}_z$ from each observer, which were 0.4, 3.0 and 3.0 degrees respectively. In 98% of cases, $\overline{V}_x$ was within 24 degrees (for Observer A which was the worst case) of V_x, showing that the LV orientation is remarkably consistent across patients. The average difference between the ground truth V_x (Observer A and B) and (i) the automatic method and (ii) the normal to the SA image planes defined by the technologist during scanning are given in Tab. 1. The magnitude of the automatic errors are very similar to the errors associated with the positioning of short axis scans during the planning of the SA slices at the MRI scanner.

(b) Position Errors. In order to compute the position errors, both the ground truth V_x and the automated V_x were intersected with the image planes and the distance between the two intersections calculated relative to the ground truth reference. The slices closest to the apex and base and the slice midway between these two are presented for the purposes of comparison. Figure 7 shows the distribution of errors for the worst case (Observer A) for the ONTARGET data. It can be seen that the automatic results and the ground truth agree closely with

Table 1. Comparison of the orientation errors (mean ± standard deviation in degrees). The parallel SA scan planes are planned to be orthogonal to V_x during image acquisition and should therefore have normals aligned with V_x.

	Automatic V_x	Normal to SA scan plane
ONTARGET Ground truth V_x Observer A	6.4 ± 4.4	6.3 ± 3.7
ONTARGET Ground truth V_x Observer B	6.1 ± 4.1	6.8 ± 4.0
ZEST Ground truth V_x Observer A	6.2 ± 4.7	6.5 ± 3.7
ZEST Ground truth V_x Observer B	5.6 ± 4.1	7.6 ± 4.7

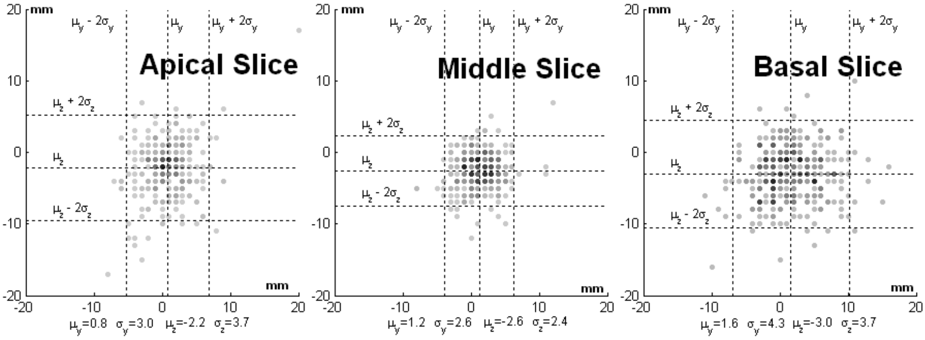


Fig. 7. Distance plots (mm) of automatic V_x relative to Observer A on apical, middle and basal slices for the ONTARGET dataset (mean and standard deviation shown for y and z directions under each plot)

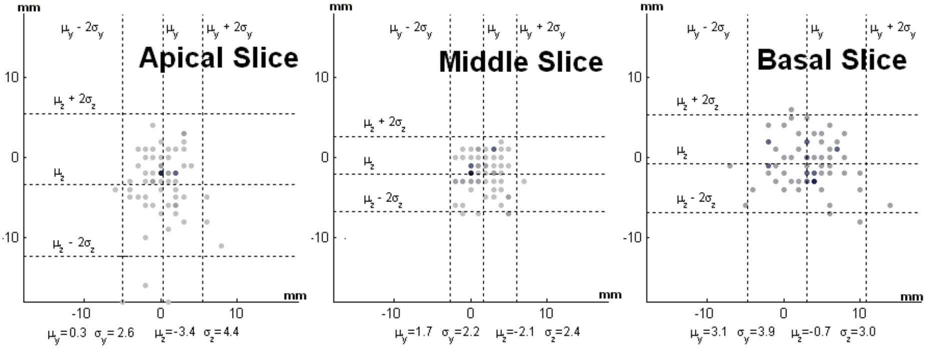


Fig. 8. Distance plots (mm) of automatic V_x relative to Observer A on apical, middle and basal slices for the ZEST dataset (mean and standard deviation shown for y and z directions under each plot)

each other. There is a small systematic bias in the V_z direction which may be caused by the conceptual differences between the manual and automatic methods (for example the ground truth V_x was measured only at end-diastole while the automatic V_x was based on images from throughout the cardiac cycle).

(b) Zest Results. As the ONTARGET dataset had been used during the development of the method, the ZEST dataset was used to provide an independent validation. The same methods were used to calculate the angular the position errors and these are also presented in Tab. 1 and in Fig. 8. In all cases the errors were similar to those from the ONTARGET dataset.

4 Conclusion

A fully automatic method of determining the position and orientation of the LV from MR images presented in this paper has been found to be both efficient and robust. The errors in the automated method are similar to those found when the orientation of the normal to the short axis scan planes are compared with LV long axis ground truth data.

References

1. Santarelli, M., Positano, V., Michelassi, C., Lombardi, M., Landini, L.: Automated cardiac MR image segmentation: theory and measurement evaluation. *Med. Eng. Phys.* **25** (2003) 149–159
2. Lorenzo-Valdés, M., Sanchez-Ortiz, G., Mohiaddin, R., Rueckert, D.: Segmentation of 4D cardiac MR images using a probabilistic atlas and the EM algorithm. In: *MICCAI'03.* (2003) 440–450
3. Mitchell, S., Bosch, J., Lelieveldt, B., van der Geest, R., Reiber, J., Sonka, M.: 3-D active appearance models: Segmentation of cardiac MR and ultrasound images. *IEEE Trans. Med. Imag.* **21**(9) (2002) 1167–1178
4. Kaus, M.R., von Berg, J., Niessen, W., Pekar, V.: Automated segmentation of the left ventricle in cardiac MRI. In: *MICCAI'03.* (2003) 432–439
5. Montillo, A., Metaxas, D., Axel, L.: Automated segmentation of the left and right ventricles in 4D cardiac SPAMM images. In: *MICCAI'02.* (2002) 620–633
6. Spreeuwers, L., Breeuwer, M.: Automatic detection of the myocardial boundaries of the right and left ventricle. *SPIE: Med. Imag.* **4322** (2001) 1207–1217
7. Danilouchkine, M., Westenberg, J., Reiber, J., Lelieveldt, B.: Accuracy of short-axis cardiac MRI automatically derived from scout acquisitions in free-breathing and breath-holding modes. *MAGMA* **18** (2005) 7–18
8. Jackson, C., Robson, M., Francis, J., Noble, J.: Automatic planning of the acquisition of cardiac MR images. In: *MICCAI'03.* (2003) 541–548
9. Anderson, C.: Rationale and design of the cardiac magnetic resonance imaging substudy of the ONTARGET trial programme. *J. Int. Med. Res.* **33**(4) (2005) 50A–57A
10. Young, A., Cowan, B., Thrupp, S., Hedley, W., Dell'Italia, L.: Left ventricular mass and volume: fast calculation with guide-point modeling on MR images. *Radiology* **216** (2000) 597–602
11. Sörgel, W., Vaerman, V.: Automatic heart localization from 4D MRI datasets. *SPIE: Med. Imag.* **3034** (1997) 333–344
12. Gering, D.: Automatic segmentation of cardiac MRI. In: *MICCAI'03.* (2003) 524–532

Two Thresholding for Deriving the Bi-level Document Image

Yu-Kumg Chen and Yi-Fan Chang

Department of Electronic Engineering
Huafan University, Taipei, Taiwan
ykchen@huafan.hfu.edu.tw

Abstract. Optical character recognition occupies a very important field in digital image processing. It is used extensively in daily life. If the given image does not have a bimodal intensity histogram, it will cause segmenting mistake easily for the previous algorithms of image binarization. In order to solve this problem, a new algorithm is proposed in this paper. The proposed algorithm uses the theory of moving average on the histogram of the fuzzy image, and then derives the better histogram. Since use only one thresholding value cannot solve this problem completely, the edge information and the window processing are introduced in this paper for advanced thresholding. Thus, a more refine bi-level image is derived and it will result in the improvement of optical character recognition. Experiments are carried out for some samples with shading to demonstrate the computational advantage of the proposed method.

1 Introduction

It is important in image processing to select a better threshold value automatically without requiring from the user to adjust a set of parameters each time when it is applied. The threshold value often extracts objects from an image. In order to get the useful information from them, to make segmentations of these images is extremely important. The bi-level image of document is to extract text from the background. To run the optical character recognition (OCR) systems properly, the OCR needs a refine bi-level image of document.

The cell-phone has already mass-produced with universal, even combine the function of the digital camera named as smart-phone in recent years. It may be combined with the functions of OCR and translation to form a special machine. With taking a document image from the special machine, the special machine can help people to translate the words of foreign language into the words of their home language in the future. However, this document image is a degraded document image with shadows, non-uniform illumination and poor quality of the source. It is relatively difficult to obtain satisfactory bi-level image.

Histogram thresholding is a well-know technique for bi-level image. In current techniques, the histogram thresholding is usually classified into tow classes, which are global thresholding and local adaptive thresholding. Global thresholding find a threshold from the information of an entire image to divide image pixels into foreground or background classes. Sahoo et al. [1] compared the performance of more than 20

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 228–237, 2006.

global thresholding methods. The comparison showed that Otsu's method [2] gave the better performance than others. Trier and Jain [3] evaluated 11 popular local thresholding methods. In their evaluations, method of Niblack [4] produced the bi-level image with highest quality. Thus, the better recognition rate can be got by using OCR for this bi-level image. With using the features of texture, Liu and Srihari [5] proposed a local thresholding method that selects an optimal threshold from a set of candidate thresholds. Solihin and Leeham's method selects an optimal threshold using histogram modified by integral ratio technique. Zhao's method [6] uses the operation of multiple window size to select a local threshold. A recent exhaustive survey of 40 image binarization methods, both global and local, is presented in Ref [7]. If the document image contains slow changing gray level of the background, local thresholding methods work well. Otherwise, it is appeared that none can be tune-up with a set of operating parameters good for all images. These techniques have been widely used in document image analysis.

In this paper, we propose a new method. The proposed method consists of three steps. In first step, with the theory of moving average [8], we modify it for the histogram of an image and derive a new algorithm named Moving-Average-Histogram for obtaining a smooth shape histogram. This smooth shape histogram contains fewer valleys in the envelope of histogram. Then in the second step, with using the general global bi-level algorithm, such as Otsu's method [2], for finding a more refine threshold value. However, just using one threshold value cannot resolve the shaded problem. With using the edge information from Sobel edge detection [9] and window processing, a two thresholding method is proposed in this paper. It can derive a more refine bi-level image. Experiments are carried out for some degraded document images that take from the smart-phone to demonstrate superior performance against four well-know techniques.

In next section, we review the best previous methods used in our experiments. Detail algorithm for deriving a smooth envelope of histogram is given in Section 3. By combining with the edge information, the proposed algorithm for obtaining the refine bi-level image is then presented in Section 4. Experimental results are shown in Section 5. Concluding remarks and potential applications are provided in the last section of this paper.

2 Compared Related Works

In this section, we review one global and three local thresholding methods that are used for the comparison and evaluation with our approach in this paper. Otsu's method [2] is a global thresholding technique to divide the histogram by selecting the threshold value to maximize the variance between the divided regions when the histogram of the two fixed points are divided with a threshold value as a standard. In Bernsen's method [10], the threshold

$$T(x,y) = (Z_{low} + Z_{high})/2$$

is used for each pixel (x, y) , where Z_{low} and Z_{high} are the lowest and highest gray levels in a square $r \times r$ neighborhood centered at (x, y). If the value $Z_{high} - Z_{low}$ is less than an another threshold l, the pixel (x, y) is set to background. Trier and Taxt [3] recommend $r = 15$ and $l = 15$.

The idea of Niblack's method [4] is to vary the threshold over the image, based on the local mean and local standard deviation. The threshold $T(x, y)$ at pixel (x, y) is calculated as

$$T(x, y) = m(x, y) + k \cdot s(x, y)$$

where $m(x, y)$ and $s(x, y)$ are the sample mean, and standard deviation values, respectively, in a local neighborhood of (x, y). The size of the neighborhood should be small enough to preserve local details, but at the same time large enough to suppress noise. Trier and Jain [3] recommend to take 15×15 neighborhood and the constant $k = -0.2$.

Sauvola and Pietikainen [11] propose a method that solves this problem by adding a hypothesis on the gray values of text and background pixels, which results in the following formula for the threshold:

$$T(x, y) = m(x, y) + (1 - k(1 - s(x, y)/R))$$

where R is the dynamics of the standard deviation fixed to 128 and k takes on positive values (usually set to 0.5). This method gives better results for document images.

3 Algorithm of Moving Average

The original images used in this paper are taken from textbook newspaper, and magazine by the smart-phone with 640×480 resolution. Let $f(x, y)$ be the original image, where x and y represent the coordinate values of each pixel in this image and their ranges are from 0 to 639 and from 0 to 479, respectively. In order to separate the text from the background, it is very important to select an optimal threshold T of gray-level. The pixel with gray-level less than or equal to T is called the character point. Pixels having a gray level lower than the threshold value T are labeled as character (black, i.e., 0 gray-level), otherwise background (whiter, i.e., 255 gray-level). Thus, the bi-level image $g(x, y)$ can be derived from

$$g(x, y) = \begin{cases} 0, & f(x, y) \leq T \\ 255, & f(x, y) > T. \end{cases}$$

A popular technique for analyzing both the overall stock market and individual stock is the theory of moving average [8] of prices, which is used to detect both the direction and the rate of change. Some number of days of closing prices is chosen for the calculation of a moving average. After initially calculating the average price, the new value for the moving average is calculated by dropping the earliest observation and adding the latest one. This process is repeated daily or weekly. The resulting moving average line supposedly represents the basic trend of stock prices. Let r be the gray-level of an image and let n_r be the number of pixel for gray-level r. Then the original histogram $h(r)$ of this image can be expressed as

$$h(r) = n_r.$$

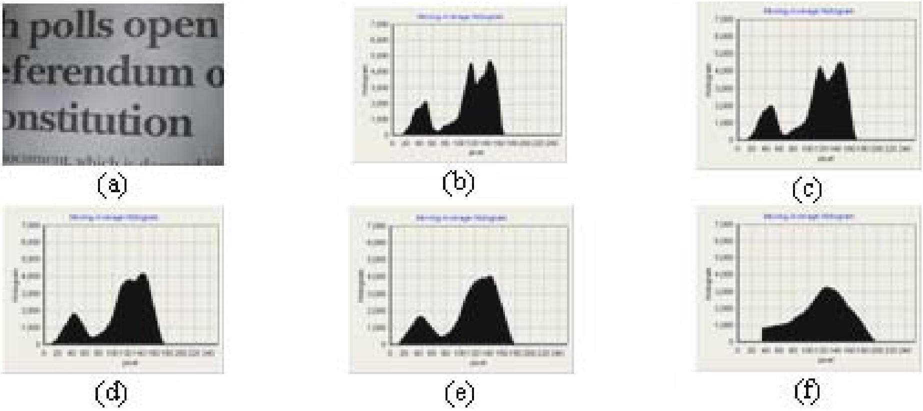


Fig. 1. Illustration of obtaining the different histograms for the propose Moving-Average-Histogram algorithm (a) original image (b) histogram with $m = 6$ (c) histogram with $m = 12$ (d) histogram with $m = 24$ (e) histogram with $m = 30$ (f) histogram with $m = 72$

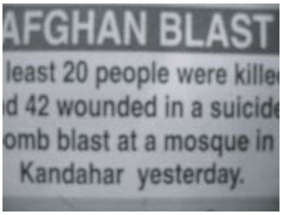
When the shape of the original histogram does not contain only one valley, the traditional thresholding methods will derive a not appropriate threshold value T. In order to erase the shape of histogram in such image with a lot of small valleys, a new algorithm named Moving-Average-Histogram is proposed in this paper based on the theory of moving average [8]. By using a window centered at gray-level r, the proposed method takes the mean value of this window to replace the original histogram $h(r)$. The new histogram $h^*(r)$ of moving average may be written as

$$h^*(r) = \frac{1}{\text{m}} \sum_{i=r-\lfloor m/2 \rfloor}^{r+\lfloor m/2 \rfloor} n_i \tag{1}$$

where m is the number of gray-levels in each window. Assuming that the range of gray-level is from 0 to L, the algorithm can be summarized as follows.

```
**Algorithm:** Moving-Average-Histogram($h(r), h^*(r)$).
**Input:** The original histogram $h(r)$.
**Output:** Moving Average histogram $h^*(r)$.
**Step 1:** Set the initial gray-level $r$ equals to $\lfloor m/2 \rfloor$.
**Step 2:** Obtain the new histogram $h^*(r)$ of Moving Average using Equation (1).
**Step 3:** Increase the gray-level $r$ and then repeat step 2 until $r$ is equal to $L - \lfloor m/2 \rfloor$.
**Step 4:** Stop.
```

However, how to determine the number of gray-level m in each window is a very important issue in this algorithm. Fig. 1 (a) shows a fuzzy image taken from a textbook. By using the proposed Moving-Average-Histogram algorithm with the different values 6, 12, 24, 30, and 72 of m, experiments shows that the value 30 of m is more suitable for the case used in this paper. Therefore, we apply m equal to 30 for each example used in the following sections of this paper.



AFGHAN BLAST
least 20 people were kille
d 42 wounded in a suicid
omb blast at a mosque in
Kandahar yesterday.

Fig. 2. An original image for testing and deriving thresholding values

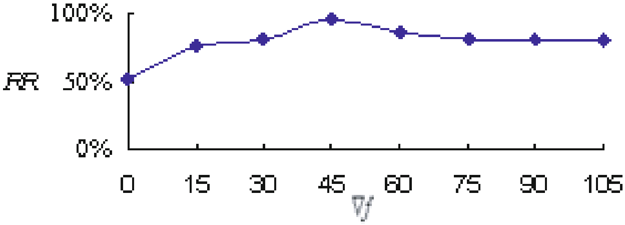


Fig. 3. Experimental results for deriving the suitable gradient $\bigtriangledown f$

The zigzag shape of the Moving Average histogram $h^*(r)$ is obtained if the small value of m is selected. Fig. 1(b) shows this case. Since there are a lot of valleys appear in this histogram, it will cause the worse bi-level threshold T gotten by using the traditional bi-level thresholding methods. On the other hand, when the large value of m is selected, the derived shape of the Moving Average histogram $h^*(r)$ is smooth, as shown in Fig. 1(f). In Fig. 1(f), there is not any valley in this histogram. Hence, this will result in the worse bi-level threshold T gotten in the next bi-level thresholding algorithm.

4 Bi-level Thresholding

In this section, we proposed a new algorithm for deriving a more refine bi-level image. There are many variables used in this algorithm. The gradient of f at coordinates (x, y) for Sobel operators [9] is represented by $\bigtriangledown f$. Let $s(x, y)$ be the image of edge information. If there is an edge at coordinates (x, y), the value of $s(x, y)$ is set to 1; otherwise, $s(x, y)$ is set to 0. We select a threshold value 45 via experimental results. If $\bigtriangledown f$ is great than 45, $s(x, y)$ is set to 1. Otherwise, $s(x, y)$ is set to 0. Let N be the total number of character in an original image and Z be the number of character which is recognized successfully by the software of OCR. Then the recognition rate RR can be expressed as

$$RR = \frac{Z}{\mathrm{N}} \times 100\%.$$

The bi-level images are tested with ABBYY FINE READER SPRINT 4 OCR software and run on a personal computer with the operating system of Microsoft Windows XP. Fig. 2 shows an original image for testing and deriving the suitable thresholding values. The recognition rate RR is plotted in Fig. 3, versus the different values of gradient$\bigtriangledown f$, such as 15, 30, 45, 60, 75, 90 and 105. Since the value 45 of the gradient $\bigtriangledown f$ can derive the best recognition rate RR, we use this value in the proposed algorithm.

We assume that the variable S is the summation of the 3×3 region of the image $s(x, y)$,is $S = \sum_{i,j=-1}^{1} s(x+i, y+j)$. The proposed bi-level thresholding algorithm can be summarized as follows.

```
Algorithm: Bi-level-Thresholding.
Input: Original image $f(x, y)$.
Output: Bi-level image $g(x, y)$.
Step 1: Get $h(r)$ and $s(x, y)$ from an original image $f(x, y)$.
Step 2: Call the procedure Moving-Average-Histogram $(h(r), h^*(r))$ and derive the
    new histogram $h^*(r)$.
Step 3: Use the general global bi-level algorithm, such as Otsu's algorithm, on the new
    histogram $h^*(r)$ to find new thresholding value $T^*$.
Step 4: Set $x$ and $y$ equal to zero.
Step 5: If $f(x, y)$ is greater than $T^*$, $g(x, y)$ is set to 255, i.e., background, and then
    go to step 8.
Step 6: If $(0 \leq f(x, y) \leq 3T^*/4)$, $g(x, y)$ is set to 0, i.e., object, and then go to step
    8.
Step 7: If $S$ is equal to 9, $g(x, y)$ is set to 0, i.e., object, otherwise, $g(x, y)$ is set to 255,
    i.e., background.
Step 8: Make the increment of $x$ or $y$.
Step 9: Repeat step 5 through step 9 until all pixels in the $f(x, y)$ are processed.
Step 10: Stop.
```

In step 3, with the experimental results, we find that the Otsu's algorithm [2] is better than the other algorithm for the proposed algorithm. Let R be the rate of S over the total number of pixels in the $B \times B$ region. It can be expressed as

$$R = \frac{S}{\mathrm{B}^2} \times 100\%.$$

With the fixed size of 3×3 region for S and original image shown in Fig. 2, the recognition rate RR is plotted in Fig. 4, versus the different rates of R, such as 0%, 11.1%, 33.3%, 55.6%, 77.8%, and 100%.

Since the rate R equal to 100%, i.e., S equal to 9, can derive the best recognition rate RR, we use this rate in the proposed algorithm. With original image in Fig. 2 and R equal to 100%, a plot of RR versus B is shown in Fig. 5. We can derive the best recognition rate RR when B is equal to 3, i.e., the 3×3 region. Therefore, we use the 3×3 region in the proposed algorithm. Let R_T^* be the T^* rate. The description of the relation RR and R_T^* are presented in Fig. 6. The rate RR can reach the

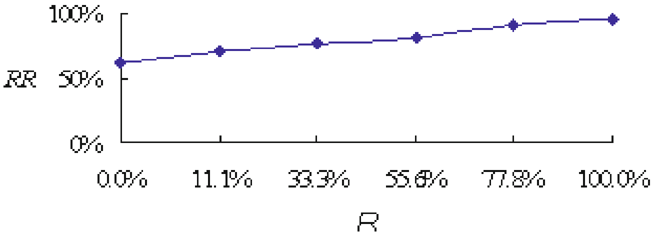


Fig. 4. Experimental results for deriving the best rate R

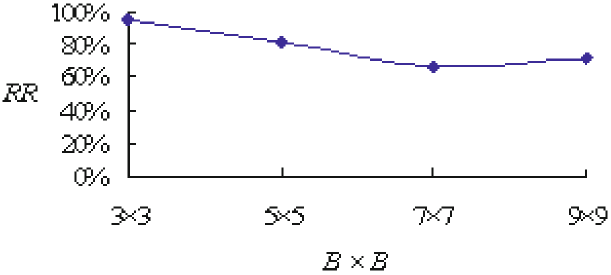


Fig. 5. Experimental results for deriving the best region size B

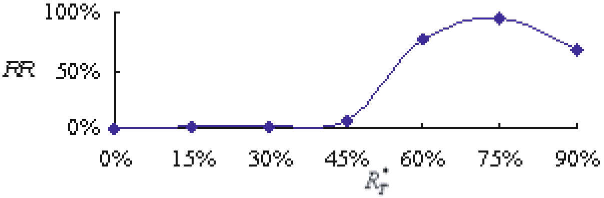


Fig. 6. Experimental results for deriving the best rate R_t^*

100% when R_T^* is equal to 75%. Therefore, the value $3/4$ in step 6 of the proposed algorithm is derived.

5 Experimental Results

The bi-level images of Bersen's algorithm, Niblack's algorithm, Sauvola's algorithm, Otsu's algorithm and our proposed algorithm are tested with ABBYY FINE READER SPRINT 4 OCR software and run on a personal computer with the operating system of Microsoft Windows XP. Some original images taken from textbooks with duskier lighting source are shown in Fig. 7. With applying all methods on the IMG1, Fig. 8 shows the results of OCR for these methods. The cyan marked words in the right column of

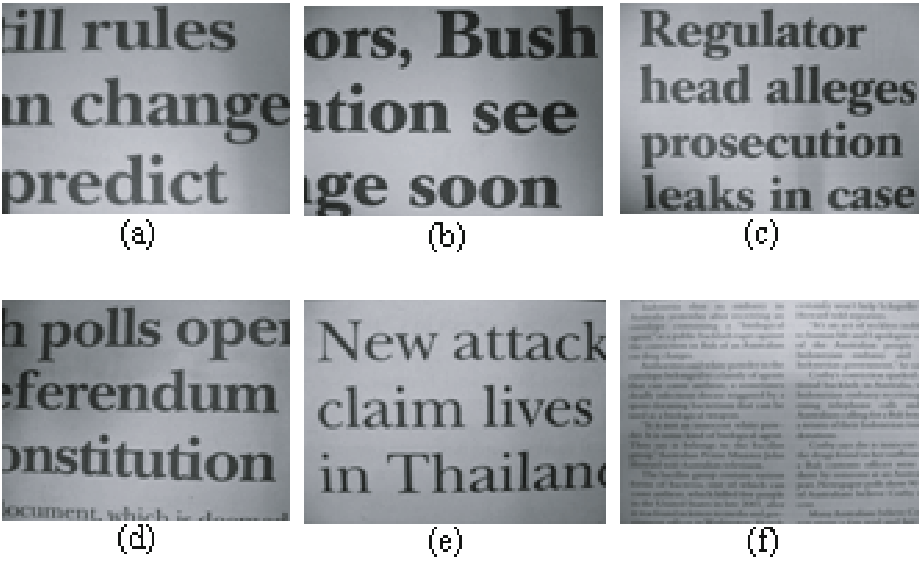


Fig. 7. The original images (a)IMG1 (b)IMG2 (c)IMG3 (d)IMG4 (e)IMG5 (f)IMG6

Table 1. The results of RR for different algorithms with different images

Methods	**IMG1**	**IMG2**	**IMG3**	**IMG4**	**IMG5**	**IMG6**
Bersen	56.5%	16.7%	28.6%	20.0%	86.7%	0.0%
Niblack	0.0%	0.0%	14.3%	37.0%	0.0%	0.0%
Sauvola	91.3%	83.3%	100%	81.5%	100%	0.0%
Otsu	86.9%	50.0%	85.7%	33.3%	83.3%	0.0%
Proposed method	100%	83.3%	100%	100%	100%	27.0%

Fig. 8 indicate the OCR errors. For example, in Bersen's method, the word ”change” is recognized as ”chaiifife”. The empty in this column of Fig. 8 represents that there is not any word or character recognized by OCR. For instance, since the bi-level image of Niblack's method generally suffers from a great amount noises of background, the OCR result is empty. Although the approach of Sauvola et al. solves the background noise problem that appears in Niblack's approach, the characters in its bi-level image become extremely thinned and broken in many cases, as shown in Fig. 8. The proposed method obtains a better bi-level image. Therefore, the proposed method has superior performance compared with all other methods and performs well even when the documents are very noisy and highly degraded.

The results of RR for different algorithms with original images in Fig. 7 are shown in Table 1. The RR values of the proposed algorithm can reach 100% for most images. Although the values of RR for the proposed algorithm are 83.3% for the IMG2 image and 27% for the IMG6 image, it can be observed that the proposed algorithm has a higher RR over the other algorithms.

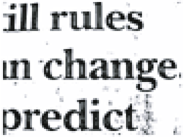
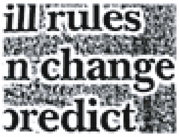
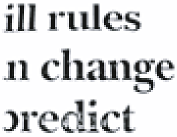
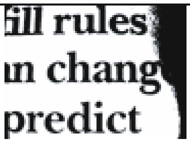
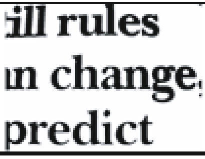
Method	Bi-level image	OCR result
Bersen		ill rules in -chaiifife
Niblack		
Sauvola		ill rules n change 3redict
Otsu		SU rules mchang predict
Proposed method		ill rules in change, predict

Fig. 8. Results of OCR for different methods

6 Conclusions

Based on the theory of moving average and edge information, this paper presents a new method for solving the global automatic thresholding problem. The proposed algorithm can derive a more refine bi-level image from an original image that is taken from the smart-phone. Since the smart-phone is not like the scanner that has light source, the quality of original image is always shaded by hand and smart-phone itself, and is always dusky. Therefore, the existing algorithms are not suitable for this case. The experimentations show that the proposed algorithm has a higher RR over the existing algorithms by Bersen, Niblack, Sauvola, and Otsu. Even the RR of the proposed algorithm can reach the 100% for some kinds of image. The advanced work of this algorithm is to merge other methods for solving more complex problems.

References

1. Sahoo, P.K., Soltani, S., Wong, A.K.C.: A survey of thresholding techniques. Computer Vision, Graphics and Image Processing **41** (1988) 233–260
2. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on systems, Man, and Cybernetics **17** (1979) 1191–1201

3. Trier, O.D., Jain, A.K.: Goal-directed evaluation of binarization methods. IEEE Transactions on Pattern Analysis And Machine Intelligence **17** (1995) 1191–1201
4. Niblack, W.: An Introduction to Digital Image Processing. Pretice-Hall, Englewood Cliffs, NJ (1986)
5. Liu, Y., Srihari, S.N.: Document image binarization based on texture features. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 533–540
6. Zhao, M., Yang, Y., Yan, H.: An adaptive thresholding method for binarization of blueprint images. Pattern Recognition Letter **21** (2000) 927–943
7. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging **13** (2004) 146–165
8. Jones, C.P.: Investments Analysis and Management. Wiley (1991)
9. Sobel, I.E.: Camera Models and Machine Perception. PhD thesis, Stanford University (1970)
10. Bersen, J.: Dynamic thresholding of gray-level images. In: Proceeding Eighth International Conference Pattern Recognition. (1986) 1251–1255
11. Sauvola, J., Pietikainen, M.: Adaptive document image binarization. Pattern Recognition **33** (2000) 225–236

A Novel Merging Criterion Incorporating Boundary Smoothness and Region Homogeneity for Image Segmentation

Zhi-Gang Tan, Xiao-Chen He, and Nelson H.C. Yung

The Department of Electrical and Electronic Engineering,
The University of Hong Kong, Hong Kong SAR, China
{zgtan, xche, nyung}@eee.hku.hk

Abstract. A novel joint region merging criterion combining region homogeneity and boundary smoothness is proposed. Previous watershed segmentation method which utilizes region homogeneity or edge integrity or both gives good results in some cases. However, for complex scenes such as images of a vehicle with irregular roadside objects reflected on the window panes, it tends to give undesired segmentation results with region boundaries not residing on real physical boundaries. Aiming at improving the segmentation of these complex scenes, we propose the incorporation of an additional measure of boundary smoothness into a new joint criterion. Based on this, an affine transform invariant measure of the smoothness of boundaries is developed, which is the equivalent width of the energy distribution function over frequencies, obtained from Fourier descriptors of the boundary. Experimental results and evaluation are presented in this paper to demonstrate the merits of the proposed method.

Keywords: watershed segmentation, merging criterion, boundary smoothness, region homogeneity.

1 Introduction

Image segmentation is a major operation in many pattern recognition/classification and image understanding applications. It is often an indispensable step before image analysis to obtain regional descriptors such that the pixel-by-pixel content in the image could be simplified, organized and better interpreted. Though hundreds of segmentation methods have been proposed in the literature, it is generally understood that the problem is ill-defined and most methods perform well under specific conditions for specific images. As for specific applications, these methods required manual tuning to output a desirable result involving particular human knowledge. For instance, watershed-based segmentation offers such a framework which allows *a priori* knowledge to bear on the algorithm [1]. Moreover, watershed segmentation has been used to integrate region and boundary information for further improvement [2].

Harris et al. [3] proposed a method which merges the most similar pair of regions at each step according to a dissimilarity function based on region homogeneity, utilizing the Region Adjacency Graph. However, as pointed out by Pavlidis and Liow

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 238 – 247, 2006.

[4], methods based only on region uniformity have the tendency to produce false boundaries because the definition of region homogeneity usually insists on a roughly constant brightness, but brightness may vary gradually within a region. Thus, it is very difficult to find homogeneity criteria which produce robust results instead of false boundaries. They also suggested that the results would be significantly improved by exploring the edge information rather than trying to fine-tune the homogeneity criteria.

With the aim of improving the segmentation results, Hernadez and Barner et al. proposed a joint region merging criterion of homogeneity and edge integrity [5]. In their method, edge integrity is measured as the ratio of the number of strong adjacent pixels (pixels with gradient value larger than a threshold) to that of boundary pixels. This quantity measures the extent of boundaries sitting on strong edges. They argued that by combining homogeneity and edge integrity together, their algorithm gives more visually appropriate segmentation. However, it is not necessary the case that strong edges are region boundaries and weak edges are not. Further, even pixels with high gradient value are not necessarily meaningful edges. This issue can be traced back to Canny's edge detector published in 1986 [6]. Therefore, simply using the strength of the edge is just not enough especially for complex scenes with cast shadows and reflections.

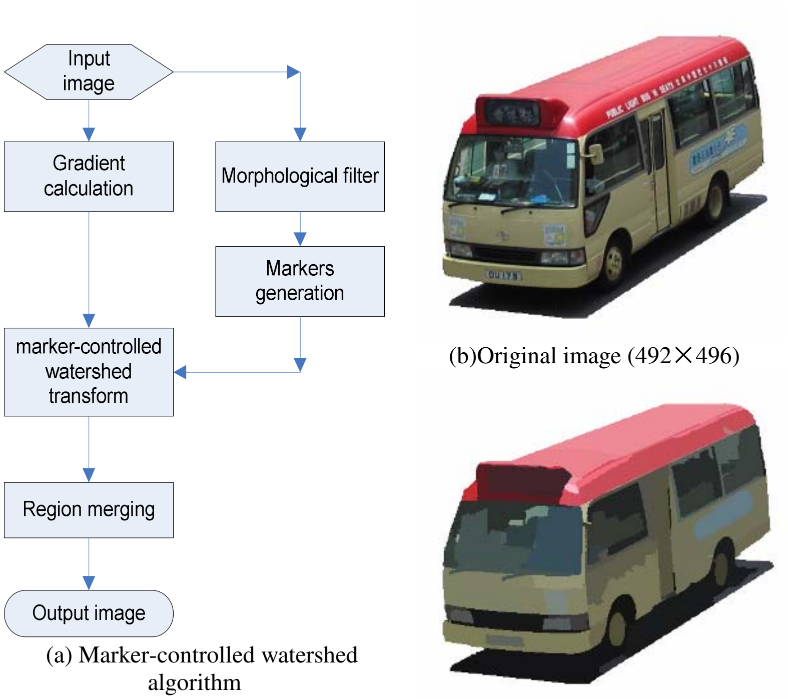
In this paper, we propose a new criterion which incorporates *a priori* knowledge of the boundaries into the merging process. It is observed that boundaries of components of many man-made objects have smooth boundaries, e.g. vehicles. As such, it is expected that the segmentation result also yields smooth boundaries. The method-active contours [7] [8]with internal balloon force pushing the contour outside along its normal is able give smooth boundaries. However, active contours need to manually initialize the starting curves. And what's more, minimizing the energy function for multiple snakes is quite time consuming. Instead, we consider a new joint criterion incorporating region homogeneity and boundary smoothness. To do this, first, a gradient image is calculated by applying a Sobel edge detector. Second, a morphological filter is applied to the original image to obtain a smoothed image. Third, markers are generated by marking the regional maxima and minima of the smoothed image. Fourth, a marker-controlled watershed algorithm is applied to the gradient image to get an initial segmentation, which reduces the number of local minima of the gradient image by allowing local gradient minima only exist inside the markers. Finally, regions are further merged to give an even more simplified yet still meaningful result by merging most similar region pairs with smallest cost of merging adjacent regions. The cost is determined by the merging criterion which calculates region homogeneity as well as smoothness of region boundaries.

In Section 2, the merging framework for over-segmentation is discussed. In Section 3, we discuss the merging criterion based on region homogeneity and boundary smoothness respectively and illustrate the problem of the criterion based only on region homogeneity. We then describe the derivation of the boundary smoothness measure and how it is incorporated into the homogeneity criteria to form a single joint merging criterion. In Section 4, we evaluate and compare the segmentation results of using the joint criterion and only the region homogeneity criterion.

2 Merging Framework for Over-Segmentation

In order to combine the map of regions (generally with false boundaries) and the map of edge outputs (generally with fine and sharp lines, but disjointed) together to give an accurate and meaningful segmentation, others have been attempted to develop methods which start with an over-segmentation result, and then merge those regions based on region homogeneity or edge integrity [2]. The methods discussed in [3],[4],[5],[9], all belong to this category, i.e. they are post-processing techniques after over-segmentation.

Our proposed method also starts with an over-segmented result produced by a morphological watershed transform of the gradient magnitude image based on immersion simulation [10]. However, the gradient operation is sensitive to noise, which results in a large number of small catchment basins not actually associated with meaningful regions. These small catchments cause the watershed transform to produce numerous negligible small regions not associated with real objects. To eliminate these extraneous local minima, we use the technique of marker-controlled watershed [11], [12] which only allows local minima occur inside the markers generated by applying an opening-by-reconstruction morphological filter to the original image and followed by identifying the region maxima and minima.



(a) Marker-controlled watershed algorithm

(b)Original image (492×496)

(c) Segmented image(108 regions)

Fig. 1. Watershed algorithm and result

The whole procedure is illustrated in Fig. 1(a). Fig. 1(c) is the initial segmented result obtained by applying the marker-controlled watershed transform to Fig. 1(b). There are 108 regions in Fig. 1(c), compared with thousands of regions in watershed transform applied directly to the gradient image. Indeed, the marker-controlled method alleviates somewhat the problem of over-segmentation. However, it should be noted that there are still a lot of regions not corresponding to physical boundaries, for instance, the regions in the windscreen and window. Therefore, further improvement is needed. This is achieved by using the Region Adjacent Graph (RAG) to analyze the relationship between the segmented regions in an image, where nodes represent adjacent regions and edge costs corresponds to a dissimilarity metric determined by the merging criteria. Once the RAG is established, the region pair with the smallest dissimilarity metric is merged. The RAG is then updated and the process repeats until a certain condition is met or terminated manually. For more details, interested reader may cross-reference [3], [5] and [13].

3 Merging Criteria

Obviously, the performance of the merging process largely depends on the merging criterion i.e. the dissimilarity metric employed. Here, we compare two different merging criteria, namely, A1, criterion based on region homogeneity, A2, the proposed merging criterion. To be consistent, the notations used here are chosen to be the same as those used in [3], [5].

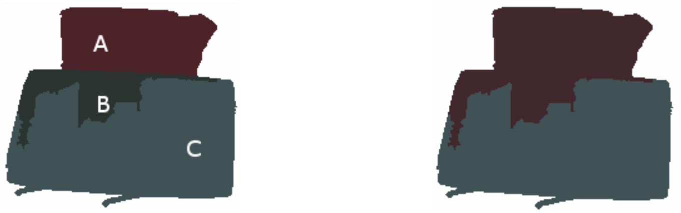
3.1 Criterion Based on Region Homogeneity

This criterion is based on similarity between their intensity levels. Harris showed in [3] that if R^*_K is the optimal K-partition that minimizes $E(R^*_K)$ then the optimal $(K\text{-}1)$-partition is obtained by merging the pair of regions with the smallest dissimilarity defined as:

$$\delta^{\mathrm{H}}(R_K^{*i}, R_K^{*j}) = \frac{\left\|R_K^{*i}\right\| \cdot \left\|R_K^{*j}\right\|}{\left\|R_K^{*i}\right\| + \left\|R_K^{*j}\right\|}[\mu(R_K^{*i}) - \mu(R_K^{*j})]^2 \tag{1}$$

where $\|\cdot\|$ denotes the number of pixels inside a region and $\mu(R_K^{*i})$ and $\mu(R_K^{*j})$ correspond to the mean value of intensity values in the adjacent regions R_K^{*i} and R_K^{*j}, respectively.

This criterion has shown some success in various applications. However, for complex scenes, it renders unaccepted results. For example, if we apply A1 to Fig 2 (a), which is the image of the front part of a minibus with the reflections on it, the dissimilarity between A and B measured by $\delta^H(A, B)$ is less than that between B and C by $\delta^H(B, C)$, which means if we want the three regions to be merged into two regions (The RGB vectors of A, B, C are (75, 34, 40), (44, 52, 50), (64, 82, 85) respectively.), A, B are to be merged first according to criterion A1, instead of B, C. The merged result is shown in Fig 2 (b), which brings about an unacceptable broken window and an irregular top panel.



(a) Regions of the vehicle front (b) Merging using A1

Fig. 2. Sample image to show the problem

3.2 A2, the Proposed Merging Criterion

The proposed criterion is an attempt to utilize boundary information as well as region homogeneity information with the aim of improving the result to a more visually appropriate segmentation. It is observed that many man-made objects have smooth boundaries other than rugged ones. So it is natural to demand the segmentation method, when applying to images of these kinds of objects, yield a result of segmented regions with smooth boundaries. However, criterion A1 does not cater for this demand. To take advantage of the boundary information, a method based on Fourier descriptors is developed to measure the smoothness of boundaries.

3.2.1 Measure of Boundary Smoothness

This section describes how the smoothness of a boundary is computed by Fourier descriptors [1]. For a boundary *l* represented as a sequence of N points $p_n = (x_n, y_n)$, *n=0,1,2,...,K-1*, the coordinates of each point can be treated as a complex number i.e. $s(n) = x(n) + jy(n)$. The Fourier descriptors of the boundary are:

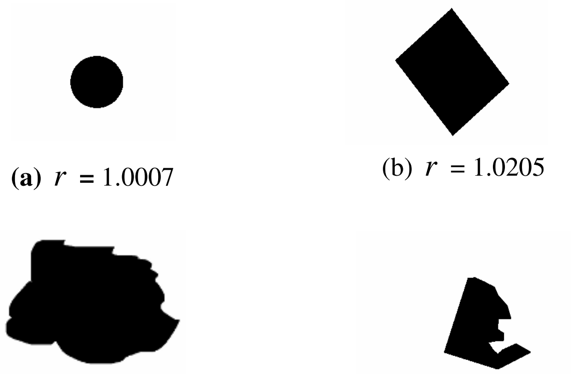
$$a(u) = \sum_{n=0}^{K-1} s(n)\, e^{-j2\pi un/K} \text{, for } u = 0,1,2,\cdots,K-1. \tag{2}$$

As we know from the Fourier transform, the coefficients *a(u)* describe the frequency components of the curve: low frequency components with u close to zero describe the general shape of an object, while the higher frequency components describe local details. It is intuitive that a rugged shape has more high frequency components and a smooth shape has less. To measure the smoothness of the curve, we could measure the concentration of the energy on the low frequencies; the more the energy concentrated on the low frequencies, the smoother the boundary is. Inspired by the equivalent width of the energy distribution function, the smoothness measure of a boundary *l {s(n) = x(n) + jy(n), n=1,2 n=0,1,2,...,K-1}* is defined as

$$r(l)=\frac{\sum_{u=0}^{K-1}\left|a^{*}(u)\right|^{2}}{a(0)}\ , \tag{3}$$

where $a^{*}(u)=\sum_{n=0}^{K-1}(s(n)-\mu)\,e^{-j2\pi un/K}$, μ is the centroid of the boundary and $\left|\bullet\right|$ denotes modulus operation.

An interesting point of this boundary smoothness measure is that it is invariant to rotation, scaling and translation. Fig 3 shows some examples of the smoothness measure. From these examples, we see that the boundary of a circle has the smallest value by the smoothness measure because a circle can be viewed as smooth in any point of its boundary. Also, we see that the more rugged the boundary, the larger value of the smoothness measure is. To draw some sense from this intuition, we call this measure shape integrity.



(a) r = 1.0007

(b) r = 1.0205

(c) r = 1.0424

(d) r =1.2694

Fig. 3. Examples of the smoothness measure

3.2.2 Incorporating the Smoothness Measure and Region Homogeneity into the Merging Criterion

After the smoothness measure is developed, the next step is to figure out the way to incorporate it into with the region homogeneity criterion. Considering two adjacent regions A and B, we have two conditions:

a) If the shape integrity of A or B is good .i.e. the $r(l^{A})$ or $r(l^{B})$ is small, which means the boundary of A is already preferred by *a priori* knowledge i.e. smooth, then the need to merge A and B is low and the cost should be large; if the shape integrity of A and B is bad, i.e. then the need to merge A and B is high and the cost should be small.

b) If the shape integrity of $A \cup B$ is good which means merging them gives a preferred shape, then the need to merge A and B is high and the cost should be small; if it is bad, then the need to merge them is low and the cost should be large.

Hence, the merging cost should be a decreasing function to the individual shape integrity of A and that of B, while being a increasing function of the shape integrity of $A \cup B$. The function used here is defined as:

$$\delta^S(A,B) = \frac{\|A\| \cdot \|B\|}{\|A\| + \|B\|} \frac{r(l^{A \cup B})^2}{r(l^A) r(l^B)} , \tag{4}$$

where l^A denotes the boundary of A and $l^{A \cup B}$ denotes the boundary of $A \cup B$.

The new criterion is implemented by combining dissimilarity metrics A1 and the measure of boundary smoothness together. Since the dynamic ranges of $\delta^{\mathrm{H}}(R_K^{*i}, R_K^{*j})$ and $\delta^S(R_K^{*i}, R_K^{*j})$ are different, we need to normalize them first before the integration. A simple but effective scheme is used here, which is to divide each of them by the maxima of $\delta^{\mathrm{H}}(R_K^{*i}, R_K^{*j})$ and $\delta^S(R_K^{*i}, R_K^{*j})$ respectively.

Finally, we have the joint merging criterion defined as following

$$\delta^m(R_K^{*i}, R_K^{*j}) = \delta_n^{\ \mathrm{H}}(R_K^{*i}, R_K^{*j}) + \alpha \cdot \delta_n^{\ S}(R_K^{*i}, R_K^{*j}) , \tag{5}$$

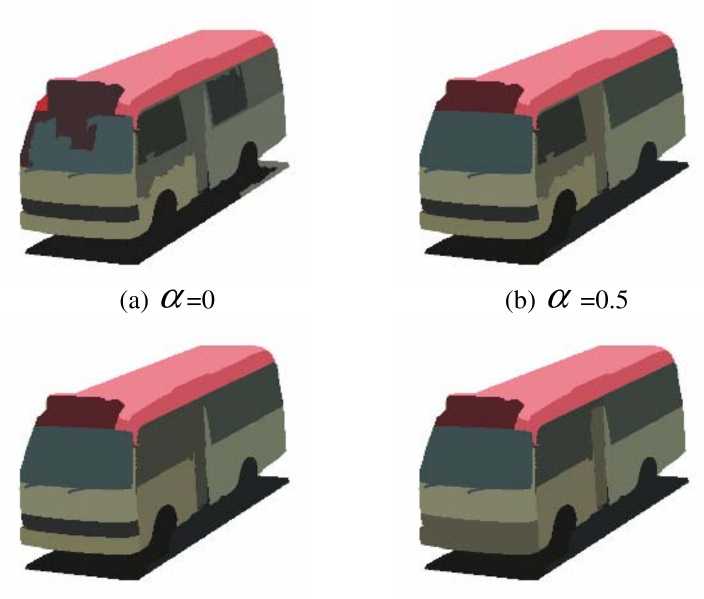
where $\delta_n^{\ \mathrm{H}}(R_K^{*i}, R_K^{*j}) = \delta^{\mathrm{H}}(R_K^{*i}, R_K^{*j}) / \max(\delta^{\mathrm{H}}(R_K^{*i}, R_K^{*j}))$,

$\delta_n^{\ S}(R_K^{*i}, R_K^{*j}) = \delta^S(R_K^{*i}, R_K^{*j}) / \max(\delta^S(R_K^{*i}, R_K^{*j}))$ and α adjusts the weight of the smoothness measure in the joint merging criterion.

4 Experiment Results and Evaluation

As we mentioned in the early part that the performance of the method largely depends on the merging criterion, here we apply our proposed method to the initial segmented image shown in Fig 1 (c) to manifest the power of the new criterion. We note that when *α = 0*, this novel criterion degrades to A1. Here, we set *α* to 0, 0.5, 0.8 and 1 respectively and apply it to Fig 1(c). The relevant segmented results are shown in Fig. 4 (a), (b), (c) and (d).

For qualitative evaluation, we see that in Fig. 4 (a), the front-top panel is merged with a part of the front window and the side window on the right hand side of the image is filled with some unwanted drippings. By observing the original image as depicted in Fig. 1(b), we find this is caused by the transparency of the window glass and its reflective ability such that other scenes around the vehicle are either projected or reflected on the image, which resulted in the failure of the regional homogeneity criterion. In Fig 4(b), the front window is no longer broken and the undesirable drippings disappear as well. Indeed, the result is more visually appropriate. As α increases, the boundaries of the segmented regions become smoother and smoother as expected, which is evident in Fig 4(c) and (d). The best value of α can be determined interactively or statistically based on large set of a certain class of images.



(a) α=0

(b) α =0.5

(c) α=0.8

(d) α=1

Fig. 4. Merged results (13 regions) using the joint merging criterion

For quantitative evaluation, Zhang concluded [14] that the empirical methods are superior to the analytical methods because no general segmentation theory exists currently. In addition, in applications like object tracking and pattern recognition in the real world, we often expect that the segmentation provides region boundaries corresponding to that of physical objects such that the boundaries are robust even when the reflected scenes on the object are changing. This is supported by the fact that information of the object is embedded in the physical boundaries rather than in the reflected scenes. Based on this fact, we determine the quality of the segmentation by overlap rate of the segmented region boundaries and the ground truth boundaries, which is defined as

$$p = \frac{\left\| l_p \cap l_s \right\|}{\left\| l_s \right\|} , \tag{6}$$

where l_p denotes the boundary pixel set of physical objects and l_s denotes the boundary pixel set of all segmented regions and $\left\| \cdot \right\|$ denotes the number of elements of

Table 1. Overlap rates for different *α*

α	Overlap rate
0	0.8048
0.5	0.8761
0.8	0.9061
1	0.9029

the set. The larger this measure is, the more the segmented region boundaries reside on the physical boundaries and the better the quality of the segmentation is.

The ground truth boundary map in Fig. 5 shows the boundaries separating the major components of the vehicle. Table 1 shows the overlap rate for different parameter values of α. As is expected with the introduction of smoothness measure, the overlap rate is improved up to 0.1013, which is 12.59% higher than the result obtained by the original regional homogeneity criterion.

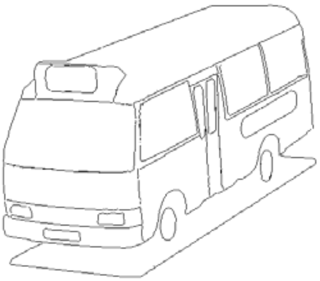


Fig. 5. Boundary reference model of vehicle

Other images of different vehicles have also been tested and showed similar results. We believe the new joint merging criterion is also suitable to other images of objects with smooth boundaries.

5 Conclusion

In order to reduce the number of regions of the segmentation yet still give a meaningful result representing the main objects in the image even when it is severely affected by irregular reflection, the proposed method employs additional knowledge of boundary smoothness of the concerned objects. The novel merging criterion combines region homogeneity and boundary smoothness in a weighted form. The smoothness measure is calculated as the equivalent width of the energy distribution function over frequencies components obtained by Fourier descriptor. This smoothness measure is invariant to rotation, scaling and translating. By setting different values of *α*, the smoothness measure exerts different weights on the merging process. The larger α, the smoother the boundary is. Appropriate value of α can be obtained interactively or statistically. Improvement of segmentation result is supported by experimental results of both qualitative and quantitative evaluation.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*: Prentice Hall, 2002.

[2] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cufi, "Yet another survey on image segmentation: Region and boundary information integration," *Computer Vision - Eccv 2002 Pt Iii*, vol. 2352, pp. 408-422, 2002.

[3] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos, "Hybrid image segmentation using watersheds and fast region merging," *Image Processing, IEEE Transactions on*, vol. 7, pp. 1684-1699, 1998.

[4] T. Pavlidis and Y. T. Liow, "Integrating Region Growing and Edge-Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 225-233, 1990.

[5] S. E. Hernandez and K. E. Barner, "Joint region merging criteria for watershed-based image segmentation," presented at Image Processing, 2000. Proceedings. 2000 International Conference on, 2000.

[6] J. Canny, "A Computational Approach to Edge-Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-698, 1986.

[7] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1988

[8] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 884-900, 1996.

[9] D. M. Wang, "A multiscale gradient algorithm for image segmentation using watersheds," *Pattern Recognition*, vol. 30, pp. 2043-2052, 1997.

[10] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, pp. 583-598, 1991.

[11] S. Beucher, "The Watershed Transformation Applied to Image Segmentation," *Scanning Microscopy Supplement*, vol. V016, pp. pp 299-314, 1992.

[12] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using Matlab*, 1st ed. Upper Saddle River, N.J.: Pearson Prentice Hall, 2004.

[13] E. H. Sergio, E. B. Kenneth, and Y. Yu, "Region merging using homogeneity and edge integrity for watershed-based image segmentation," *Optical Engineering*, vol. 44, pp. 017004, 2005.

[14] Y. J. Zhang, "A survey on evaluation methods for image segmentation," *Pattern Recognition*, vol. 29, pp. 1335-1346, 1996.

A New Local-Feature Framework for Scale-Invariant Detection of Partially Occluded Objects

Andrzej Sluzek[1,2]

[1] Nanyang Technological University, School of Computer Engineering
Blk N4, Nanyang Avenue, Singapore 639798
assluzek@ntu.edu.sg
[2] SWPS, ul. Chodakowska 19/31, Warszawa, Poland

Abstract. Partially occluded objects are typically detected using local features (also known as interest points, keypoints, etc.). The major problem of the local-feature approach is the scale-invariance. If the objects have to be detected in arbitrary scales, either computationally complex methods of scale-space (multi-scale approach) are used, or the actual scale is estimated using additional mechanisms. The paper proposes a new type of local features (keypoints) that can be used for scale-invariant detection of known objects in analyzed images. Keypoints are defined as locations at which selected moment-based parameters are consistent over a wide radius of circular patches around the keypoint. Although the database of known objects is built using the multi-scale approach, analyzed images are processed using only a single-scale. The paper focuses on the keypoint building and matching only. Higher-level issues of hypotheses building and verification (regarding the presence of known objects) are only briefly mentioned.

1 Introduction

Detection of known objects in observed scenes is considered one of the fundamental tasks in machine vision. The task becomes more difficult when the objects are only partially visible. In such cases, the generally accepted approach is to identify objects from their local visual characteristics which remain unchanged (at least some of them) even if the object itself is partially occluded.

The idea of using local features (keypoints, local visual saliencies, interest points, characteristic points, corner points – several almost equivalent names exist) in image analysis can be traced back to the 80's (e.g. [1], [2]). Although initially stereovision and motion tracking were the most typical applications, it was later found that the same approach can be used in more challenging tasks (e.g. matching images and detection of partially occluded objects). A well-known Harris-Plessey operator (e.g. [1]) was combined with local descriptors of detected points to solve general object recognition problems in which local features from analyzed images are matched against a database of images depicting known objects (e.g. [3], [4]). The intention was to perform recognition of arbitrarily rotated objects under partial occlusions.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 248 – 257, 2006.

The published results indicated that three issues are fundamental for successful applications of the proposed algorithms, namely: *illumination changes*, *perspective distortions* and *scale changes*.

Illumination changes (photometric transformations) may both affect repeatability of keypoint detection and distort their descriptors values. To achieve repeatability of keypoint detection under illumination changes, several acceptable solutions have been proposed (e.g. [5]). The keypoint descriptors sensitivity to illumination variations can be usually handled using the following typical approaches: (a) invariants moments (e.g. [6], [7]) for intensity-based descriptors, or (b) normalization techniques (e.g. [8]) for gradient-based descriptors.

Perspective distortions are generally approximated by affine transformations. Since none of the existing algorithms is fully affine (the initial steps of keypoint detection and description are generally performed with no consideration to perspective effects) only relatively minor distortions are typically assumed (e.g. [9], [8]). Stronger affine distortions are ignored and the database 3D objects are modeled using just multiple views taken from various viewpoints (differing typically by 15-30 degrees).

The *scale-invariance* problem (an illustrative example is given in Fig.1) is more difficult. The existing solutions of this problem are far from effective. Generally, to achieve scale invariance of local features either computationally expensive scale-space approaches are attempted (e.g. [8], [10]) or the appropriate scale is estimated using additional means (e.g. [5], [11]). So far, no method is known that can scale-invariantly match local features using just a one-size window scanning the images captured in arbitrarily changing scales.

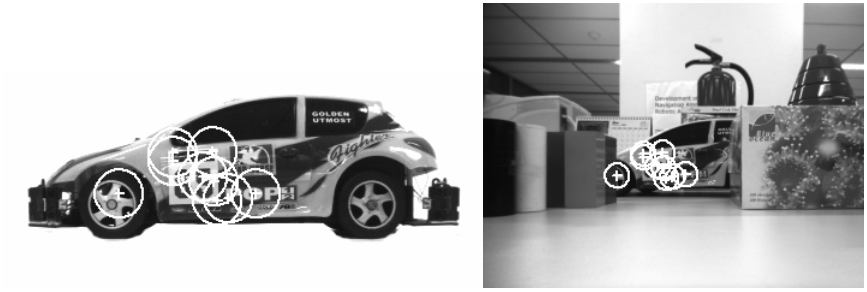


Fig. 1. Correctly selected scales for matching local features in both images (from [5])

In this paper we propose a method that handles the scale-invariance issues in a novel way. We attempt to detect known objects in acquired images using fixed-scale local features, even though the images might be captured in a wide range of scales.

The central idea of the proposed method is a new type of local features (keypoints). The keypoints are built and described using approximations of the surrounding circular patches. The descriptors of primary importance are angular parameters (obtained from locally computed moments) of the approximations. A multi-scale approach is employed in the database keypoint building operation, the approximarions are built for circular patches of varying size. A keypoint is identified if the descriptors are uniform over a significantly wide range of patch diameters. This process may be computationally intensive. However, the object detection algorithm (analysis of the input images) employs only a single scaled, i.e. the images scanned using only a one-size window. Therefore, the proposed method is suitable for typical applications where the model-building operations can be performed offline (and time and/or computational constraints do not exist) while the object detection task is to be performed online (possibly with tight time constraints).

The principles of keypoint building are presented in Section 2. In Section 3, we discuss how keypoints are detected in incoming images and matched with the database keypoints. Due to limited size of this paper, we focus only on detection and matching mechanisms, while higher-level issues of hypotheses building and verification (about presence of a known object in the analyzed image) are only briefly mentioned. Section 4 concludes the paper.

2 Pattern-Based Keypoints

Our previous paper [12] proposed a method for approximating circular images using predefined patterns. Corners and corner-like patterns (e.g. junctions) are particularly important as they generally preserve their geometry over a wide range of geometric/photometric transformations and over various radii of circular patches. Thus, in this work we focus on two most popular selected corner-like patterns, i.e. proper corners and T-junctions.

The model configuration of a corner over a circle of radius ***R*** is defined by two angles and two intensities as shown in Fig.2A. Similarly, the model configuration of a T-junction consists of two angles and three intensities (Fig.2B).

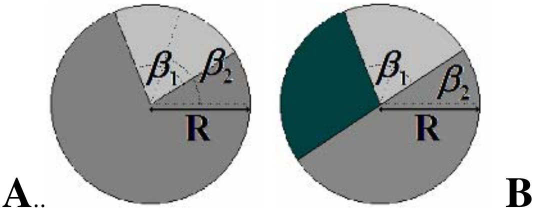


Fig. 2. Model configurations of a corner (A) and a T-junction (B)

Given any circular image of radius ***R***, the parameters of its optimum corner approximation can be found using moment-based expression specified in [12]. The orientation angle β_2 (see Fig.2A) is detrmined by

$$\beta_2 = \arctan 2(\pm m_{01}, \pm m_{10}) \tag{1}$$

while the angular width β_1 is computed from

$$\beta_1 = 2\arcsin\sqrt{1 - \frac{16[(m_{20} - m_{02})^2 + 4m_{11}^2]}{9R^2\left(m_{10}^2 + m_{01}^2\right)}} \text{ or } 2\arccos\frac{4}{3}\sqrt{\frac{(m_{20} - m_{02})^2 + 4m_{11}^2}{R^2\left(m_{10}^2 + m_{01}^2\right)}} \tag{2}$$

For T-junctions (Fig.2B) β_1 angular width and β_2 orientation angle can found from

$$\frac{\pi}{2}-\beta_2-\frac{\beta_1}{2}=\frac{\arctan 2(\pm m_{02}\mp m_{20},\pm 2m_{11})}{2} \tag{3}$$

$$m_{01}\cos\beta_2-m_{10}\sin\beta_2=\pm\frac{4}{3R}\sqrt{(m_{20}-m_{02})^2+4m_{11}{}^2} \tag{4}$$

The intensities of the optimum approximations can be also estimated using moment-based expressions (see [12]). Exemplary circular windows (containing both actual corners and T-junctions, as well as more random contents) and their optimum approximations are shown in Fig.3.

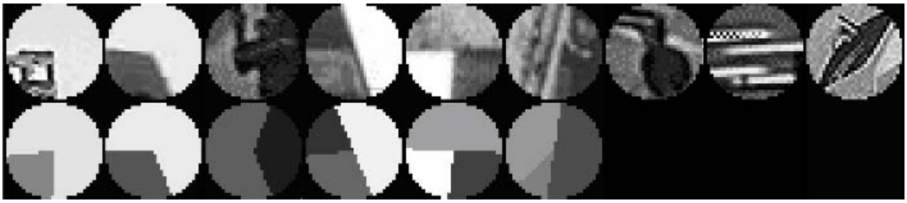


Fig. 3. Examples of circular images and their optimum corner or T-junction approximations

Straightforward calculations prove that results produced by Eqs (1)-(4) are invariant to linear illumination changes and that the angular width β_1 is invariant under any similarity transformation. Extensive experiments have also shown that the results are stable (unlike, for example, the corner approximation proposed in [13]) under both high- and low-frequency noise, image texturization and partial over-and under-saturation of image intensities. As shown in Fig.3, for some irregular images the approximations may not exist, i.e. the corresponding equations have no solutions.

It seems, therefore, that corner and T-junction approximations of circular windows are good candidates for invariant local features. However, generally they are not scale-invariant, i.e. re-scaling an image (without the corresponding change of the window size) may dramatically change the visual content of the window and thus its approximations (see Fig.4 for a corner approximation example).

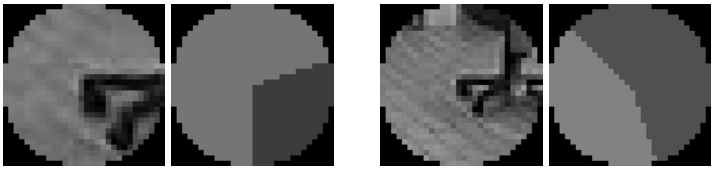


Fig. 4. Changes of corner approximations under image rescaling

Nevertheless, images of objects of interest may contain locations where geometry of the approximations does not change with image rescaling (or with the corresponding rescaling of circular windows). We propose to use such locations as candidates for scale-invariant keypoints. Fig.5 shows examples of such locations is a complex scene.

More formally, we identify a pattern-based keypoint candidate (currently only corner and T-junction patterns are used) in a model image at a location where for the window radius $\boldsymbol{R}$ ranging from $\boldsymbol{R}_{\mathbf{min}}$ to $\boldsymbol{R}_{\mathbf{max}}$, the angular width β_1 and the orientation angle β_2 are approximately the same, i.e. the approximations are consistent over a certain range of scales.

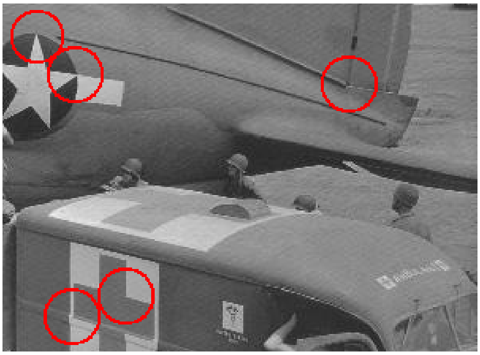


Fig. 5. Exemplary locations at which corner and T-junction approximations are expected to be stable over a wide range of circular windows

Fig.6 shows an exemplary circular window and the variations of its approximations for $\boldsymbol{R}$ ranging from 3 to 40 pixels. This image is actually a corner so that its corner approximation is visually more appealing. However, the T-junction approximation (though it has a less straightforward visual interpretation) can be created as well.

The results given in Fig.6 indicate that any circular window of $\boldsymbol{R}$ between approx. 14 and 31 pixels would be equally representative for this image fragment. The contents of minimum-size and maximum-size sub-windows are also shown for reference in Fig.6 next to the original window.

If the same image fragment is present in another scene (even if scaled, rotated and photometrically distorted) we can use the scanning window of any radius between $k\boldsymbol{R}_{\mathbf{min}}$ and $k\boldsymbol{R}_{\mathbf{max}}$ (where k represents the relative scale between the model image and the analyzed scene) and the match between both fragments would be found.

Fig.7 shows an attempt to identify a corner-based keypoint for another randomly selected image fragment. In this case, there is no uniformity in the obtained corner approximations, and for many values of $\boldsymbol{R}$ the solution for β_2 does not exist (the solution for orientation angle β_1 almost always exists). Thus, this image fragment cannot be considered a keypoint candidate.

Thus, the database of known objects images would be built using the keypoints selected from candidates that have stable approximations over a sufficiently wide range of radii. Additionally, keypoints with extreme angular widths (i.e. close to either 0 or 180 degrees) may be excluded, see [12]. A database entry for a single image of an object (note that objects are typically represented by multiply views) would also include angular specifications of the selected approximation-based keypoints, and additionally the geometric relations between the keypoints (for example, in the form of *shape-graph* proposed in [5]). The acceptable range of $\boldsymbol{R}$ may be memorized as well.

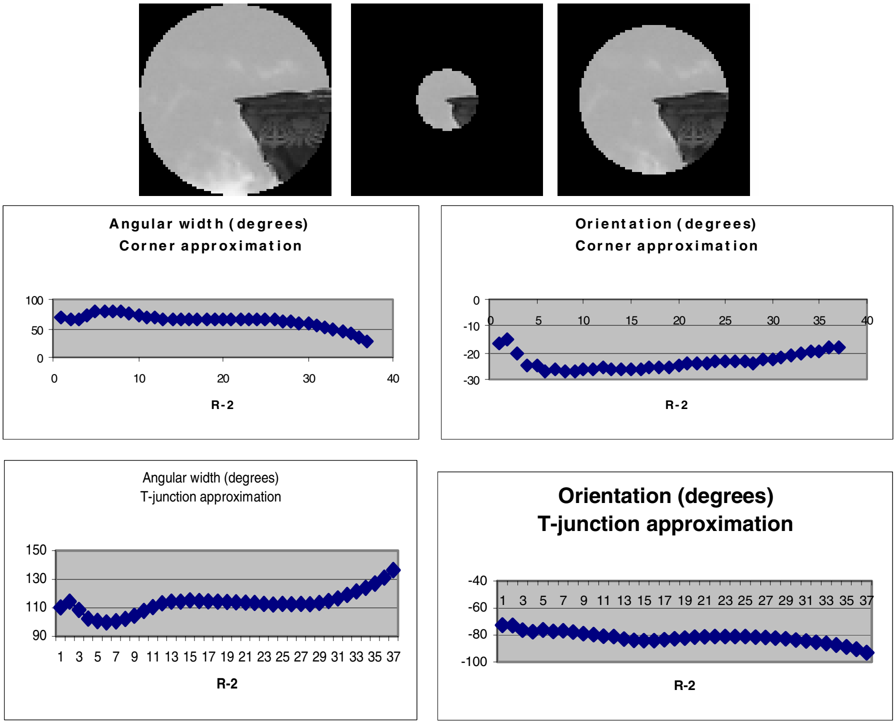


Fig. 6. Exemplary image fragment and its approximations over a wide range of radii

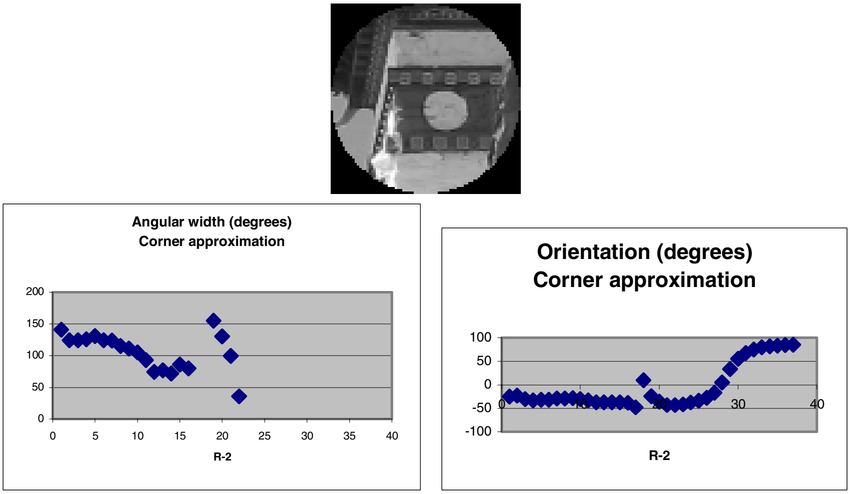


Fig. 7. Unsuccessful attempted search for a corner-based keypoint candidate

For selected applications, further rules can be applied in the process of database keypoints selection. For example, additional descriptors may be used and their stability over a range of radii (or other regular behaviours) might be required. Usually, those secondary descriptors should be invariant to image rotations and intensity variations. The following simple moment-based expressions are examples of descriptors that are invariant (for circular images) under similarity transformations and linear illumination transformations (see also [14]):

$$\frac{(m_{20}-m_{02})^2+4m_{11}^2}{R^2\left(m_{10}^2+m_{01}^2\right)} \qquad \frac{2(m_{20}+m_{02})-R^2 m_{00}}{R\sqrt{m_{10}^2+m_{01}^2}} \tag{5}$$

Because the above expressions are applied to circular windows only, they can be much simpler than more general invariants (proposed for colour images and areas of arbitrary shapes) presented in [7].

It should be noted that because of a more restrictive process of database keypoint building, the proposed method would produce fewer keypoint than the alternative algorithms (e.g. [4], [5] or [8]). This fact can be seen as an advantage (lower complexity of matching operations) but detection abilities under major occlusions and/or in cluttered scenes may be affected. If a too small part of an object of interest is seen, it may just contain too few keypoints of the proposed type to detect the presence of the object.

3 Principles of Keypoint Matching and Object Detection

The intended application of the proposed keypoints is detection of database objects (including partially occluded objects) in either robotic applications (navigation, visual surveillance) or in visual data mining problems. We assume that objects should be detected scale-invariantly (at least within a certain range of scales) even though the processed images are scanned using only a fixed-size circular window.

For any location of the scanning window, its content is approximated by corners and T-junctions. A candidate match for a database keypoint is identified if both the corner approximation and the T-junction approximation have the angular widths β_2 in close enough to the corresponding database angular widths. This simple selection technique may lead to "*very_many_to_one*" matches, but the experiments have shown that combination of two approximations (plus additional criteria) significantly reduces ambiguous matches. To further reduce the number of ambiguities, we propose to use a sub-window (the recommended radius of the sub-window is ~60% of the window's radius) for which the same operations are performed. If the sub-window approximations are different than their window-based counterparts, the location is not considered.

Fig.8 shows two exemplary test images in which matches for Fig.6 keypoint are searched for. It should be noted that the database keypoint and both images are in different scales each, yet the size of the scanning window is the same in both test images. The test images are additionally photometrically distorted and one of them is rotated. Fig.9 presents the candidate matches detected using only the corner

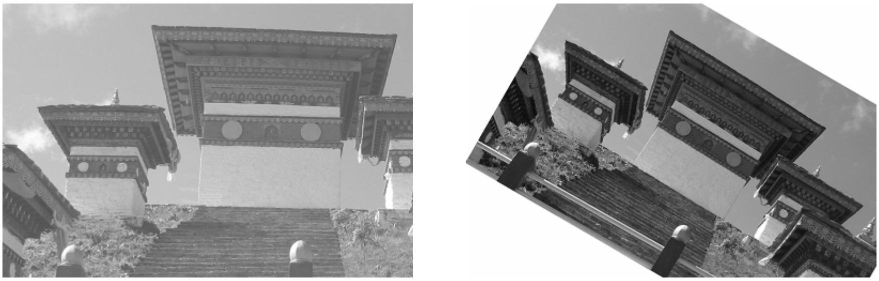


Fig. 8. Test images where matches to Fig.6 keypoint are detected

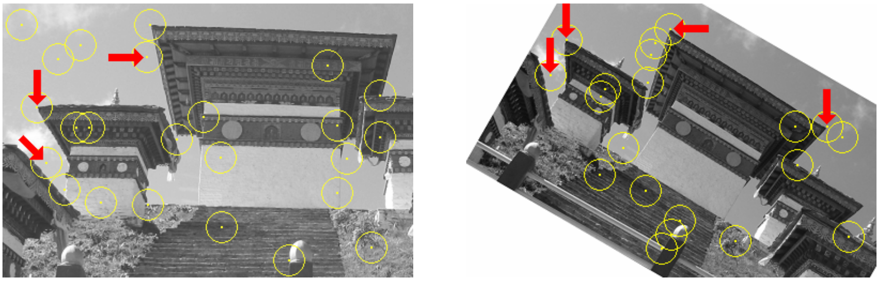


Fig. 9. Candidate matches to the keypoint of Fig.6. Radii of the scanning window are 15 and 10 pixels (window and its sub-window, respectively). Only the corner approximations are used.

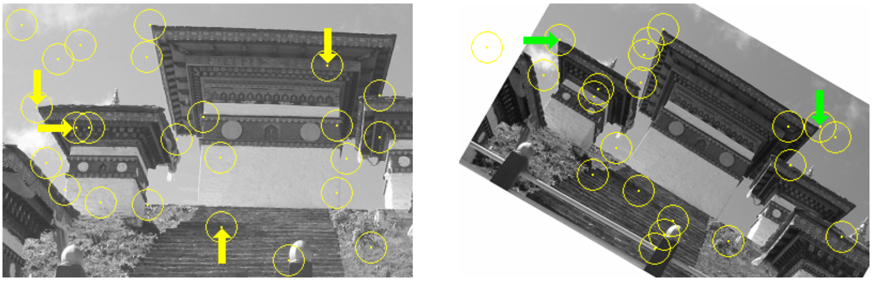


Fig. 10. Candidate matches to the keypoint of Fig.6 obtained using the corner approximations and confirmed by the T-junction approximations

approximations. The number of candidates is quite large, but it can be dramatically reduced if additional rules are added to the keypoint specification. In this case we use two straightforward facts: "*the acute part of the corner is darker*", and "*the contrast between both parts of the corner should exceed a threshold value*"). After the rules have been applied, only very few candidates (pointed by arrows in Fig.9) are selected as matches to the keypoint of interest.

If T-junction approximations are incorporated, the number of matching candidates is even further reduced. Fig.10 shows candidates (pointed by arrows) obtained by the corner approximations, and additionally confirmed by the T-junction approximations. The intersection of choices shown in Fig.9 and Fig.10 finally produces only a very small number of potential matches (one and two, correspondingly).

If processed images contain candidates matching several database keypoints, the problem of ambiguous matches can be further solved by comparing the orientation angles β_1. Even if the object is rotated in the acquired image, the values of β_1 should be consistently rotated for all candidates matching keypoints from the same database object. Thus, with at least two keypoints visible in the image, the ambiguities can be usually solved. If at least three keypoints from the same database object image are consistently matched with the test image, hypotheses can be generated not only about the presence of the object but also about its relative scale.

A framework for efficient hypotheses generation/verification for problems with hundreds or thousands keypoints in the database (and correspondingly large numbers of candidates in the acquired images) is presented in [5]. We believe, however, that in many typical applications (a search for a particular object in large collections of images, for example) less sophisticated mechanisms based on the principles briefly described above are sufficient.

4 Summary

In this paper we present principles and preliminary exemplary results of a novel technique for scale-invariant detection of known objects in acquired images. Unlike other scale-invariant techniques, our method is using only a single scale for scanning analyzed images. However, images of database (known) objects are processed with multiple scales in order to identify (and characterize) keypoints that are invariant under a sufficiently wide range of scales. Thus, matching a database keypoint to the candidate keypoints extracted from incoming images can be done for various image-scales (as long as the scanning scale is within the scale range of the matching keypoint).

The proposed keypoints are significantly different from typical gradient-based keypoints used in the existing alternative techniques. Our keypoints are based of moment-derived pattern approximations of circular patches around keypoints (currently only two patterns, i.e. corners and T-junctions, are used). Their primary descriptors (the angular width and orientation of the approximations) are robust under illumination changes, noise, texturization, and other typical real-world effects.

The paper presents fundamentals of keypoint-building and keypoint-matching. Higher-level issues of object detection are not discussed. At higher levels, we intend to use approaches already presented in previously published papers.

In the future work we plan to apply the methodology in two typical tasks: (1) visual search and/or surveillance in autonomous robotic systems and (2) data mining in large collections of visual information. As an important step towards even higher efficiency of the method, an FPGA-accelerator for the low-level image analysis operations (i.e. corner approximation) is planned. It would be prospectively used in both tasks.

References

1. Harris, C., Stephens, M.: A combined corner and edge detector. Proc. 4th Alvey Vision Conference, Manchester (1988) 147–151.
2. Moravec, H.: Rover visual obstacle avoidance. Proc. Int. Joint Conf. on Artificial Intelligence, Vancouver (1981) 785–790.
3. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. IEEE Transactions on PAMI, Vol.19 (1997) 530–534.
4. Zhang, Z., Deriche, R., Faugeras, O., Luong, Q.T.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artificial Intelligence, Vol.78 (1995) 87–119.
5. Islam, M.S.: Relative scale method to recognize and localize objects for robotic applications. PhD thesis, NTU (SCE), Singapore (2005).
6. Maitra, S.: Moment invariants. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, New York (1979) 697–699.
7. Mindru, F., Tuytelaars, T., Van Gool, L., Moons, Th.: Moment invariants for recognition under changing viewpoint and illumination. Computer Vision & Image Understanding, Vol.94 (2004) 3-27.
8. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. Journal of Computer Vision, Vol.60 (2004) 91-110.
9. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. Proc. European Conf. on Computer Vision ECCV-2002, Copenhagen (2002) 128–142.
10. Lindeberg, T.: Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. Int. Journal of Computer Vision, Vol.11 (1993) 283–318.
11. Sluzek, A., Islam, M.S.: Visual target detection in unstructured environments – a novel technique for robotic navigation. Robot Design, Dynamics, and Control (CISM Courses and Lectures No.487 – eds T.Zielinska, C.Zielinski), Springer, Wien New York (2006) 431-438.
12. Sluzek A.: On moments-based local operators for detecting image patterns. Image & Vision Computing. Vol.23 (2005) 287-298.
13. Rosin P.L.: Measuring corner properties. Computer Vision & Image Understanding, Vol.73 (1999) 291-307.
14. Sluzek, A., Islam, M.S., Palaniappan, A.: Using interest points for visual detection and identification of objects in complex scenes. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems IROS-2006, Beijing (2006) - accepted.

Color Correction System Using a Color Compensation Chart for the Images from Digital Camera

Seok-Han Lee, Sang-Won Um, and Jong-Soo Choi

Dept. of Image Engineering, Graduate School of Advanced Imaging Science, Multimedia,
and Film, Chung-Ang University, 221 Huksuk-Dong, Dongjak-Ku,
156-756, Seoul, Republic of Korea
{ichthus, sangwon, jschoi}@imagelab.cau.ac.kr
http://imagelab.cau.ac.kr

Abstract. In this paper, we describe the color correction system using a color compensation chart for the color images from digital cameras. Most of the conventional methods for the color corrections of images are based on the spectral analysis for the captured images. The proposed system introduces a different approach for the color correction. That is, the color correction is performed by dynamic tone reproduction and direct transformation between the captured colors and the reference colors. The color correction process consists of two steps, i.e., the profile creation process and the profile application process. During the profile creation process, the relationships between the captured colors and the reference colors are estimated. And the system creates a color profile and embeds the estimation result in the profile. During the profile application process, the colors in the images which are captured under the same condition as that of the chart image are reproduced using the created color profile. To evaluate the performance of the system, we perform experiments under various conditions. And we compare the results with those of widely used commercial applications.

1 Introduction

With the advance of digital imaging devices such as digital cameras, their abilities to represent accurate colors have been important issues. However, accurate color correction is a difficult problem for digital cameras, because color images of unknown objects are captured under various unknown conditions such as illuminations. Furthermore, color distributions of captured images are also dominated by characteristics of the cameras. Accordingly, colors in a digital image may be different from the actual ones taken into the image. This color distortion can be a critical problem for some industrial fields using color images. Most of the conventional approaches are based on the analysis of spectral reflectance and image formation models for the captured images [1, 2, 3, 4, 5, 6]. However, many of them are focused on limited conditions, and rarely provide the examples for practical implementations.

In this paper, we describe the color correction system using a color compensation chart for digital images captured by digital camera. The proposed system introduces a different approach from the conventional techniques. That is, the color correction is performed by the estimation of dynamic tone reproduction curve (TRC) and the direct transformation between the captured colors and the reference colors. Therefore, the

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 258 – 269, 2006.

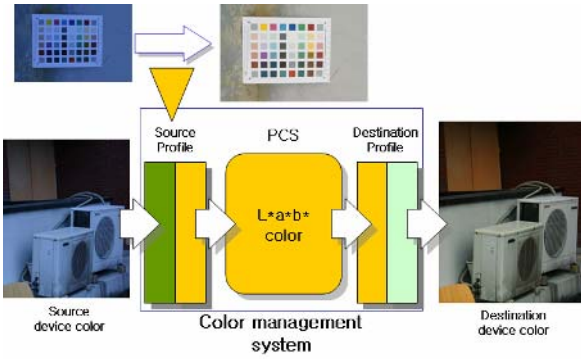
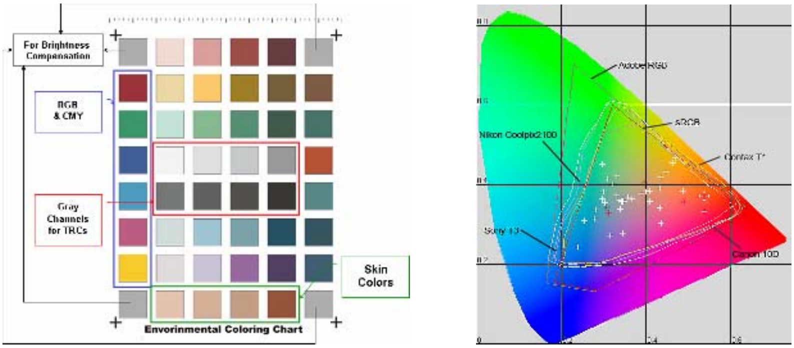


Fig. 1. Block diagram of the proposed system

color correction is achieved without the conventional image formation model which is hardly solved perfectly. We employ the concepts of color management system (CMS) and profile connection space (PCS) to realize the proposed system. CMS and PCS have been defined by ICC to overcome the inter-device color consistency problems, and are supported by most of digital imaging or displaying devices [10]. Fig.1 shows the block diagram of the proposed system. First, the image of the color compensation chart is captured. And the relationship between the colors of the patches in the image and those of the internal reference chart is estimated. Then, the system creates an ICC color profile, and embeds the relationship in the profile. The created profile is used as the source profile as in Fig.1 to reproduce the colors of the other images which are captured under the same condition as that of the chart image. To evaluate the performance of the system, we perform experiments under various conditions. And we compare the results with those of widely used commercial systems.



(a) The color compensation chart

(b) Color gamuts considered to determine the patch colors

Fig. 2. The color compensation chart and its gamut

This paper is consists as follows. In section 2, we describe the color compensation chart designed for the proposed system, and the color correction process is discussed

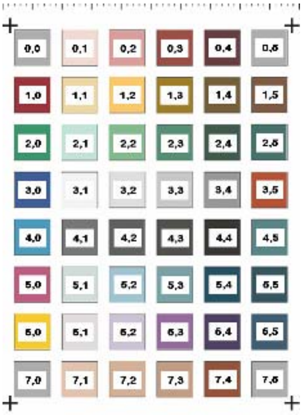
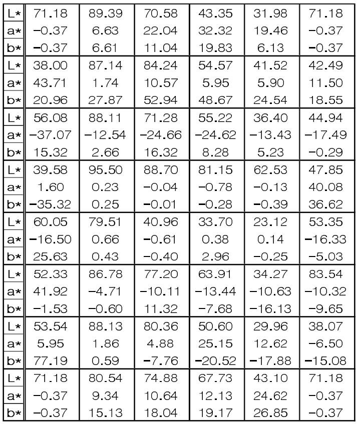


Fig. 3. Indexes of the patches



L*	71.18	89.39	70.58	43.35	31.98	71.18
a*	−0.37	6.63	22.04	32.32	19.46	−0.37
b*	−0.37	6.61	11.04	19.83	6.13	−0.37
L*	38.00	87.14	84.24	54.57	41.52	42.49
a*	43.71	1.74	10.57	5.95	5.90	11.50
b*	20.96	27.87	52.94	48.67	24.54	18.55
L*	56.08	88.11	71.28	55.22	36.40	44.94
a*	−37.07	−12.54	−24.66	−24.62	−13.43	−17.49
b*	15.32	2.66	16.32	8.28	5.23	−0.29
L*	39.58	95.50	88.70	81.15	62.53	47.85
a*	1.60	0.23	−0.04	−0.78	−0.13	40.08
b*	−35.32	0.25	−0.01	−0.28	−0.39	36.62
L*	60.05	79.51	40.96	33.70	23.12	53.35
a*	−16.50	0.66	−0.61	0.38	0.14	−16.33
b*	25.63	0.43	−0.40	2.96	−0.25	−5.03
L*	52.33	86.78	77.20	63.91	34.27	83.54
a*	41.92	−4.71	−10.11	−13.44	−10.63	−10.32
b*	−1.53	−0.60	11.32	−7.68	−16.13	−9.65
L*	53.54	88.13	80.36	50.60	29.96	38.07
a*	5.95	1.86	4.88	25.15	12.62	−6.50
b*	77.19	0.59	−7.76	−20.52	−17.88	−15.08
L*	71.18	80.54	74.88	67.73	43.10	71.18
a*	−0.37	9.34	10.64	12.13	24.62	−0.37
b*	−0.37	15.13	18.04	19.17	26.85	−0.37

Fig. 4. L*a*b* values of the patches

in section 3. The experimental results are given in section 4, and the conclusions are drawn in section 5.

2 Color Compensation Chart for the Proposed System

For the proposed system, we designed the color compensation chart as shown in Fig.2(a). This chart provides the reference colors for the estimation of the relationship between the colors in the image and the actual colors. Four patches which have gray color are located in the four corners of the chart to compensate the brightness of every patch. Eight patches located in the center area of the chart are used to create the tone reproduction curves. The patch colors were determined base on the conditions such as general colors observed in the most common environment, general skin colors of Korean people [11, 12]. The color gamuts of widely used digital cameras were also considered, because the proposed system is for the images captured by digital cameras. Fig.2(b) shows the color gamuts of the compensation chart and considered color spaces. The '+' marks located in the considered gamuts in Fig.2(b) denote the patch colors. Fig.3 shows the index of every patch which is used in the entire process, and Fig.4 presents the L*a*b* values of the patches.

3 Color Correction Algorithm for the Proposed System

The color correction algorithm for the proposed system consists of the profile creation process and the profile application process. During the profile creation process, the relationships between the captured colors and the internal reference colors are estimated. Then the system creates the color profile, and embeds the estimation result in the color profile. In the profile application process, the colors in the image which is captured under the same condition as that of the chart image are reproduced using the created color profile.

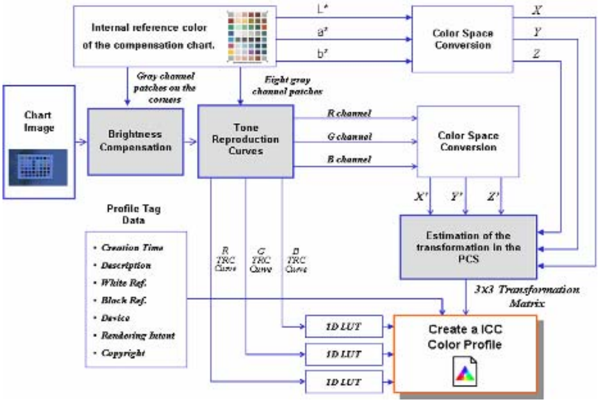


Fig. 5. Profile creation process

3.1 Profile Creation Process

Fig.5 shows the block diagram of the color profile creation process. The first step of the profile creation process is the brightness compensation, which is to make the brightness of every patch uniform. After the brightness compensation, the TRCs are estimated, and the color correction matrix is computed in the PCS. The color correction matrix maps the input colors to the internal reference colors, and the colors of the image is corrected by this matrix during the profile application process. Finally, the system creates the ICC color profile, and embeds the estimation results in the profile.

3.1.1 Brightness Compensation of the Chart

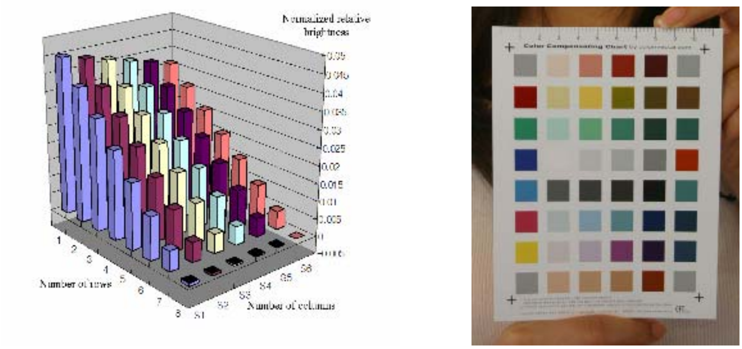
Due to some conditions such as shadows or illuminations, the brightness of every patch in the chart may not be uniform. If the brightness of every patch is not uniform, the entire estimation for the profile creation process may be led to inaccurate results. Therefore, we compensate the brightness of every patch using the colors of the four patches located on the four corners of the chart (i.e., (0, 0), (7, 0), (0, 5), and (7, 5) in Fig.3). For the brightness compensation, every patch color is converted into HSI color, and the normalized intensity (i.e., ‘I’ value) is used as the brightness value. Before the brightness compensation, relative brightness values of the four patches should be estimated to produce the brightness map which consists of the relative brightness values of the patches in the image. For the relative brightness estimation, the minimum of the brightness difference between the four patches in the image and those in the internal reference chart is estimated, and the brightness values of the four patches in the image are adjusted using the minimum brightness value as follows:

$$\begin{aligned} &D_S = \underset{P_{(i,j)}}{Min} \left| P_{(i,j)} - P'_{(i,j)} \right|, \\ &P_{S(i,j)} = P_{(i,j)} - D_S, \quad i = 0, N_h\text{-}1, N_v\text{-}1, \quad j = 0, N_h\text{-}1, N_v\text{-}1, \end{aligned} \tag{1}$$

where $P(i, j)$ is the brightness of the patch in the image, and $P'(i, j)$ means the brightness of the patch in the internal reference chart. D_S is the smallest brightness difference, and N_h and N_v mean the number of patches in the horizontal and the vertical direction respectively. And i and j are used to denote the indexes of the four patches as shown in Fig.3. By Eq.(1), we get the relative brightness values $P_S(i, j)$ for the four patches in the image. Based on the relative brightness values, the brightness compensation is achieved using bilinear interpolation [13]. That is, relative brightness of every patch is interpolated linearly in the vertical and the horizontal direction using the relative brightness differences of the four patchs estimated by Eq.(1) as follows:

$$\begin{aligned} f_{(i,0)} &= \frac{i}{N_v - 1} P_{S(N_v-1,0)} + \frac{N_v - 1 - i}{N_v - 1} P_{S(0,0)}\,, \\ f_{(i,5)} &= \frac{i}{N_v - 1} P_{S(N_v-1,5)} + \frac{N_v - 1 - i}{N_v - 1} P_{S(0,N_h-1)}, \quad i = 0,1,2,\ldots,N_v - 1, \end{aligned} \tag{2}$$

$$g(i, j) = \frac{N_h - 1 - j}{N_h - 1} f_{(i,0)} + \frac{j}{N_h - 1} f_{(i,N_h-1)}, \quad j = 0, 1, 2, \ldots, N_h - 1, \tag{3}$$



(a) Brightness map for the compensation chart

(b) The image of the compensation chart

Fig. 6. The image of the color compensation chart and its brightness map

where i and j denote the indexes of the patches as in Fig.3. $g(i, j)$ is the interpolated brightness. By Eq.(2) and Eq.(3), we get the brightness map for the color compensation chart in the image. The relative brightness value of the brightness map is subtracted from every patch's brightness in the image to equalize the brightness of every patch. Fig.6(a) shows the brightness map for the chart image of Fig.6(b). The relative brightness value of Fig.6(a) is subtracted from the brightness value of every patch in Fig.6(b) to make the brightness value of every patches in the image uniform. After the brightness compensations, the color of every patch is converted back into RGB color for the next process.

Table 1. Regression errors

Index	Red Channel	Green Channel	Blue Channel
(3, 1)	0.004876	0.003311	0.001765
(3, 2)	0.008573	0.010219	0.005493
(3, 3)	0.007408	0.003542	0.002145
(3, 4)	0.007514	0.011883	0.010104
(4, 1)	0.012203	0.016832	0.005139
(4, 2)	0.007819	0.037835	0.089622
(4, 3)	0.008355	0.026986	0.039552
(4, 4)	0.027958	0.014997	0.037026
Mean Absolute Error	0.010588	0.015701	0.023856

Table 2. ICC profile Tags for the system

Tag	Signature	Data Type
mediaWhitePointTag	wtpt	XYZType
mediaBlackPointTag	bkpt	XYZType
CopyrightTag	cprt	multiLocalizedUnicodeTag
profileDescriptionTag	desc	multiLocalizedUnicodeTag
redMatrixCouilmnTag	rXYZ	XYZType
greenMatrixCouilmnTag	gXYZ	XYZType
blueMatrixCouilmnTag	bXYZ	XYZType
redTRCTag	rTRC	curveType
greenTRCTag	gTRC	curveType
blueTRCTag	bTRC	curveType

3.1.2 Estimation of the Tone Reproduction Curves

Based on the chart images, the proposed system estimates the TRCs dynamically. The eight patches located in the center area of the chart are use for the TRC estimation process (i.e., from (3, 1) to (3, 4) and from (4, 1) to (4, 4) in Fig.3). For each of R, G, and B channel, the function to fit the color values of these eight patches in the image to those in the internal reference chart is estimated. This function is used not only as the TRC, but also as the white balancing function, because the function fits the input colors to the ideal gray colors. This process is performed in R, G, and B channels independently. In this paper, we apply curve regression to fit the color values. We suppose that the function $g(x)$ for the curve regression is given as follows [13]:

$$\begin{gathered} g(x) = a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x) + a_4 f_4(x),\ 0 \le x \le 1, \\ f_1(x) = 1,\ f_2(x) = x,\ f_3(x) = \sin(x),\ f_4(x) = \exp(x), \end{gathered} \tag{4}$$

where a_n is undetermined coefficient, and x is normalized color in the range of 0.0 to 1.0. The coefficients of Eq.(4) are determined to minimize the sum of the square of the differences between the color values of the eight patches in the image and those of the internal reference chart as follows:

$$R = \sum_{i=1}^{8} [r_i]^2 = \sum_{i=1}^{8} \left[y_i - \sum_{n=1}^{4} a_n f_n(x_i) \right]^2 . \tag{5}$$

Fig.7(a) is the result of the curve regression, and Fig.7(b) represents three TRCs for each of R, G, and B channel. The regression errors are shown in Table 1. From Fig.7(a) and the table, we can verify that the estimation process does not produce any critical error. The TRCs are preserved in the ICC profile as described in section 3.1.5.

3.1.3 Color Space Conversion

As the system adopts PCS and CMS, RGB colors of the patches in the image should be converted into those of XYZ color space before the estimation of the transformation. As defined in the specification ICC.1:2004-10, the PCS is relative to the illuminant D50 [10]. Moreover, the connection between the PCS and the device color space should be considered. Therefore, following conditions are considered for the color conversion.

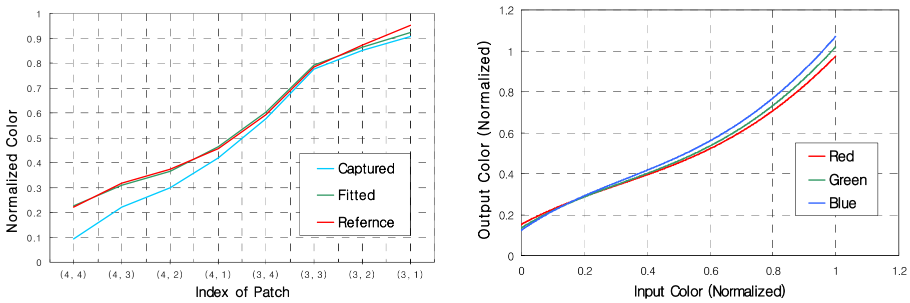
1) **Reference white of the PCS:** The PCS defined in the ICC specification is based on the illuminant D50. Therefore, the reference white point in the PCS should be equal to D50 (i.e., X=0.9642, Y=1.0, Z=0.8249).

2) **Device color space:** We employed sRGB as the device color space. It is defined based on the illuminant D65.

3) **Chromatic adaptation transformation:** As the illuminant of the device color space is not equal to that of the PCS, the chromatic adaptation method is required for the color space conversion. We used the linear Bradford model for the chromatic adaptation transformation [9, 10].

By the condition 1), 2) and 3), we get the color space conversion matrix employed for the system as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{M}_{adapt}\mathbf{M}_{CVS}\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.436052 & 0.385082 & 0.143087 \\ 0.222492 & 0.716886 & 0.060621 \\ 0.013929 & 0.097097 & 0.714185 \end{bmatrix}\begin{bmatrix} R \\ G \\ B \end{bmatrix}, \qquad (6)$$



(a) Result of curve regression

(b) TRCs for each of R, G, and B channel

Fig. 7. Results of the TRC estimation process

where, $\mathbf{M}_{adapt}$ denotes the chromatic adaptation transformation matrix defined by the linear Bradford model, and $\mathbf{M}_{CVS}$ is the color space conversion matrix from the PCS to sRGB color space.

3.1.4 Correction of the Image Color in the PCS

The colors of the image are corrected in the PCS by the linear transformation which maps the colors of the patches in the image into those of the reference colors. The transformation is estimated as follows:

$$\mathbf{M}_{XYZ} = \mathbf{M}_{Crr}\mathbf{M'}_{XYZ} = \begin{bmatrix} X_1 & X_2 & & X_N \\ Y_1 & Y_2 & \cdots & Y_N \\ Z_1 & Z_2 & & Z_N \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} X'_1 & X'_2 & & X'_N \\ Y'_1 & Y'_2 & \cdots & Y'_N \\ Z'_1 & Z'_2 & & Z'_N \end{bmatrix}, \tag{7}$$

where $\mathbf{M}_{Crr}$ is the transformation matrix which maps the patch colors of the image into those of the reference colors. Eq.(7) is estimated by SVD.

3.1.5 The ICC Color Profile Created by the System

The estimated relationship for the color correction is embedded in the ICC color profile. Table 2 shows the tags used for the ICC profile created by the system. They are defined in the specification ICC.1:2004-10 [10].

1) mediaWhitePointTag, and mediaBlackPointTag: The reference white and the reference black are embedded in each tag. As the chromatic adaptation transformation is applied to the color space conversion matrix in the proposed system, the mediaWhitePointTag is set to the values of D50. The mediaBlackPointTag is set to the reference color of black (X = 0, Y = 0, Z = 0).

2) redMatrixColumnTag, greenMatrixColumnTag, and blueMatrixColumnTag: These tags are intended to be used for the transform from the device color space to the PCS. In the proposed system, the color space conversion matrix of Eq.(6) multiplied by the color correction matrix $\mathbf{M}_{Crr}$ of Eq.(7) is embedded in these tags.

3) rTRCTag, gTRCTag, and bTRCTag: The TRCs are preserved in each of rTRCTag, gTRCTag, and bTRCTag. The proposed system employs the curveType data type defined in the ICC specification. For the curveType, a 1-D lookup table (LUT) is established to map the input color to the output color as follows:

$$redTRC[i] = g_R(x_i),\ greenTRC[i] = g_G(x_i), blueTRC[i] = g_B(x_i), i = 0,1,..,N_S, x_i = \frac{i}{N_S}. \tag{8}$$

Here N_S denotes the number of samples of $g(x)$. In the proposed system, $g(x)$ is sampled at 1024 points in the range of 0.0 to 1.0.

3.2 Profile Application Process

The profile application process is the inverse process of the profile creation process. Fig.8 shows the block diagram of the color correction process [10].

4 Experimental Results

The experiments are performed in two categories. First, the performance of the system is evaluated under the various light sources. We use SpectraLight III by GretagMacbeth which can produce standard light sources defined by CIE. The second experiment is performed for the white balance distortions. The white balance of the

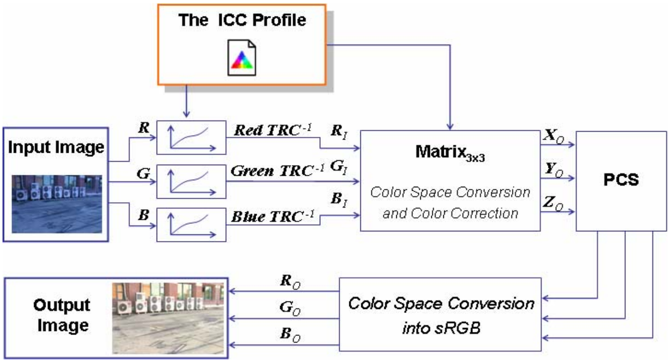


Fig. 8. Color correction process

captured image is distorted using the preset values included in the digital camera. The performance of the system is evaluated by the average of the color difference of every patch. The color difference of every patch is estimated as follows [9]:

$$\Delta E^*_{ab} = ((\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2)^{1/2} \ ,$$
$$\Delta L^* = L^*_1 - L^*_2 , \ \Delta a^* = a^*_1 - a^*_2 , \ \Delta b^* = b^*_1 - b^*_2 \ , \qquad (9)$$

where L^*_1, a^*_1, and b^*_1 represent the captured color, and L^*_2, a^*_2, and b^*_2 mean the internal reference color. Then, the color difference of every patch is averaged, and we consider this average of the color difference as the color correction error. The performance of the system for each experiment is compared with those of three commercial applications: “ProfileMaker 5” and “i1” by GretagMacbeth, “QP Color Kit 1” by QP Card. The images are captured using Canon EOS-10D digital camera.

4.1 Experimental Results for Various Illuminations

For the experiments, we consider the CIE standard illumination models generated by SpectraLight III: D65 (6500K), CWF(4150K), Horizon(2300K), U30(3100K), and Standard light “A”(2856K). Under these illumination models, the color differences are estimated in the CIELAB space. We employ “ColorChecker” color chart by GretagMacbeth as the test chart, which is not used by any systems. The experiment is performed as follows. First, the image of each system’s native color chart is captured under the illumination model. And each system’s color profile is created for the color correction. Then, the image of the test chart is captured under the same illumination model as that of the native color chart image. The image of the test chart is corrected using the color profile created by each system. Finally, the color correction error of each system is estimated. Fig.9 shows the images of the experimental results. The color correction errors for Fig.9 are shown in Table 3, in which each value denotes the average of the color difference of every patch in the test chart. From the results, it seems that ProfileMaker 5 shows the best performance. However, Table 3 represents that there is not much difference between the correction errors of the proposed system and those of the other systems. Moreover, the proposed system produces superior resultsto some systems under the illuminations such as “Horizon”, “U30”, and “A”. Therefore, we can

Table 3. The color differences for the illumination models

	D65	CWF	Horizon	U30	A
Original	4.24	8.1	28.4	17.6	21.9
Proposed	5.82	6.91	11.7	8.01	8.15
QP card	4.32	5.38	17.6	7.32	7.92
Profile Maker 5	3.44	4.42	9.57	6.23	6.88
i1	3.86	4.77	12.4	8.75	9.05

Table 4. Comparison of the color differences for the white balance values

	Sunlight	Shadow	Cloud	Tungsten	Fluorescent
Original	6.94	16.82	9.39	31.2	22.41
Proposed	5.87	5.95	6.02	5.99	6.21
QP card	8.44	8.72	8.57	11.71	11.52
Profile Maker 5	5.95	6.44	7.68	15.53	14.21
i1	6.13	6.89	7.45	16.29	15.82

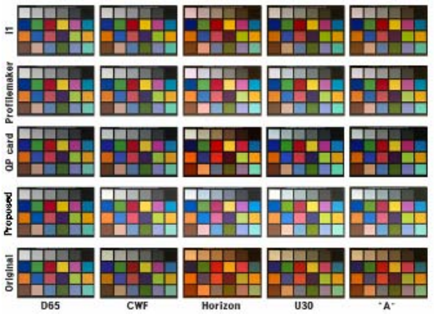


Fig. 9. Comparison of the performances under the various illumination models

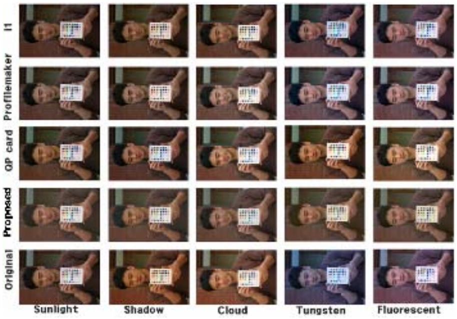


Fig. 10. Comparison for the white balance distortions

conclude that the proposed system shows almost equivalent performance to the other commercial systems under the various illuminations.

4.2 Experimental Results for the White Balance Distortions

The white balance of the image to be captured is distorted using the preset values in the digital camera as follows: 1) "Sunlight"(5200K), 2) "Shadow"(7000K), 3) "Cloud"(6000K), 4) "Tungsten"(3200K), and 5) "Florescent"(4000K). The experiment is performed in the similar manner as that of section 5.1. However, the white balance value is distorted instead of the illumination variation. Table 4 presents the color differences for the white balance distortions. We can verify that the proposed system shows almost steady results removing the effect of the white balance distortions, whereas the other systems are largely affected by the white balance distortions. Moreover, it shows the superior performance to the other systems for every preset values. Fig.10 shows the results for the white balance distortions.

4.3 Experimental Results for Arbitrary Conditions

Fig.11 shows the experimental results for the arbitrary conditions, which is to ensure the visibility presented to the users. Fig.11(a) and Fig.11(b) show the performance for the white balance distortion. We can verify that the colors are corrected so as to be almost equasl to the original ones. The images of Fig.11(c) present the results for the outdoor scenes. The corrected images show great enhancement. The enhancement is also verified by the blue tones of the sky in the image. Fig.11(d) also confirms that the color correction performance of the proposed system.

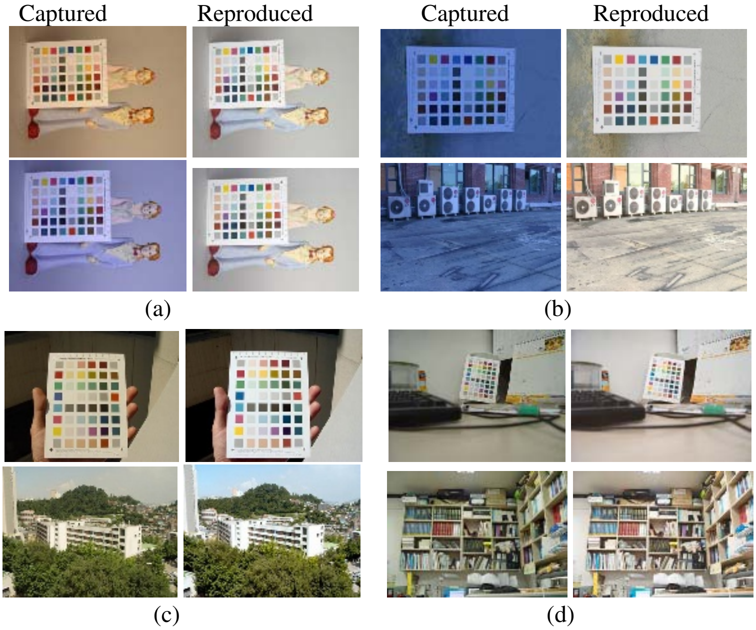


Fig. 11. Experimental results under the arbitrary conditions

5 Conclusion

We proposed the color correction system, and verified that the system shows satisfactory performance. However, we found some problems. First, the refinement process to reduce the residual error should be considered. As the transformation to correct the colors in the XYZ space was supposed to be linear, there may exists some residual error. This might be reduced by the refinement process. And the effect of the noise should be considered to produce more reliable quality.

Acknowledgments. This work is supported by the Ministry of Education and Human Resource Development under BK21 project, and financially supported by the Ministry of Education and Human Resource Development(MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor(MOLAB) through the fostering project of the Lab and Excellency.

References

1. G. Finlayson, “Color in Perspective,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 10, pp. 1034-1038, Oct. 1996
2. K. Barnard, V. Cardei, and B. Funt, “A Comparison of Computational Color Consistency Algorithm-Part-I: Methodology and Experiments with Synthesized Data,” *IEEE Transactions on Image Processing*, Vol. 11, No. 9, pp. 972-984, Sep. 2002

3. P. C. Hung, “Colorimetric Calibration in Electronic Imaging Devices using a look-up-table Model and Interpolation,” *Journal of Electronic Imaging*, Vol.36, No.1, pp. 53-61, Jan. 1993.
4. H. S Chen and H. Kotera, “Three-dimensional Gamut Mapping Method Based on the Concept of Image Dependence,” *Journal of Image Science and Technology*, Vol.46, No.1, pp44-52, Jan. /Feb. 2002.
5. C. Münzenmayer, D. Paulus, T. Wittenberg, “A Spectral Color Correction Framework for Medical Applications,” *IEEE Transactions on Biomedical Engineering*, Vol.53, No.2, pp. 254 – 265, Feb. 2006.
6. Y. Komia, K. Ohsawa, Y. Ohya, T. Obi, M. Yamaguchi, and N. Ohyama, “Natural color reproduction system for telemedicine and its application to digital camera,” *IEEE International Conference on Image Processing 1999*, pp50-54, Oct., 1999.
7. E.J. Lee, “Favorite Color Reproduction for Reference Color,” *IEEE Transactions on Consumer Electronics*, pp10-15, Feb., 1998.
8. D.H. Kim, H.C. Do, S.I. Chien, “Preferred Skin Color Reproduction Based On Adaptive Affine Transformation,” *IEEE Transactions on Consumer Electronics*, Vol.51, No.1, Feb., 2005.
9. B. T. Ahn, E. B. Moon, and K. S. Song, “Study of Skin Colors of Korean Women”, *Proceedings of SPIE*, Vol.4421, pp705-708, June 2002.
10. J. S. Jun, “A Study on the Color Scheme of Urban Landscape Based on Digital Image Color Analysis,” *Yonsei Graduate School of Human Environmental Sciences*, 2002.
11. H. C. Lee, “Introduction to Color Imaging Science,” *Cambridge Univ. Press*, 2005
12. Specification ICC.1:2004-10, *International Color Consortium*, 2004.
13. S. Nakamura, “Applied Numerical Methods in C,” *Prentice Hall*, 1995.

A Comparison of Similarity Measures for 2D Rigid MR Image Registration Using Wavelet Transform

Shutao Li, Shengchu Deng, and Jinglin Peng

College of Electrical and Information Engineering, Hunan University,
Changsha, 410082, China
shutao_li@hnu.cn, michael_dun@hotmail.com, pengjl2005@126.com

Abstract. Medical image registration is a decisive step in medical image processsing. In intensity-based image registration methods, multiresolution coarse-to-fine strategy is often used to speed up the registration process. In this paper, several commonly-used similarity measures were compared under multiresolution wavelet framework. The similarity measures are energy, joint entropy, mutual information, normalized mutual information, correlation ratio, and partitioned intensity uniformity. Experimental results give a guidance to the selection of appropriate similarity measures for registration in a multiresolution wavelet framework.

1 Introduction

Medical image registration is to eliminate the difference with translation, rotation, distortion, and locate the corresponding anatomical point couples at the same spatial location between different medical images. As an essential part of medical image processing, medical image registration has emerged as a particularly active field [1-3].

For image registration problems, the proper selection of similarity measures is decisive for the quality of registration. Conventionally, medical image registration criteria fall into three major categories: landmark-based, segmentation-based, and intensity-based. Intensity-based registration avoids the complexity of segmentation or salient point extraction, thus being widely used in recent years. Unfortunately, this kind of methods also brings heavy computation load at the same time. Multiresolution is often used to speed up the registration process, in which registration is done in a coarse-to-fine framework [4]. In other words, rough estimations are found using sub-sampled images and the fine-tuning of the solution is implemented at higher resolution. This can be done with either Gaussian pyramids or pyramids constructed from wavelet decomposition [4-7].

For most of the existed methods, one similarity criterion was used to match the images at all multiresolution levels. However, the images are varied in characteristics at different resolutions, thus it should be more reasonable to use different matching criteria in different resolution levels.

In this paper, we compare six intensity-based registration measures under multi-resolution wavelet framework. For each resolution level, the six different metrics are

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 270 – 279, 2006.

compared on the decomposed approximation image to find out which one is the most accurate and robust. This will result in the findings of the most appropriate similarity measures for registration in multiresolution wavelet framework.

The rest of paper is organized as follows. The discrete wavelet transform and its applications to image registration are briefly introduced in section 2. Similarity measures used in this paper are given in section 3. Section 4 reports the experimental results. Finally, some concluding remarks are given in section 5.

2 Discrete Wavelet Transform and Image Registration

2.1 Discrete Wavelet Transform(DWT)

Wavelet transform represents functions as a superimposition of wavelets which can be written as:

$$\psi^{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \tag{1}$$

They are dilated and translated versions of a mother wavelet ψ. While extended to 2-D, it could be regarded as configuration of a bunch of high-pass filters and low-pass filters. The general idea can be represented in diagram as follow:

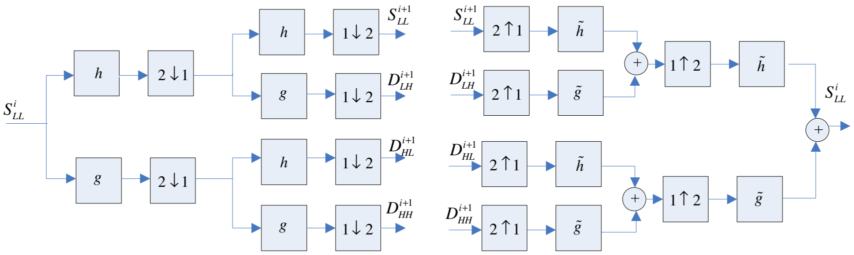


Fig. 1. One stage of 2D DWT and inverse DWT. $2\downarrow 1$ denotes keeping one column out of two, and $1\downarrow 2$ denotes keeping one row out of two. $2\uparrow 1$ denotes putting one column of zeros between each column and $1\uparrow 2$ denotes putting one row of zeros between each row. g and h are high-pass and low-pass analysis filters respectively, while $\tilde{g}$ and $\tilde{h}$ are their synthesis counterpart.

As can be seen from the diagram, 2-D signal at level i will be decomposed into four sub-band images D^{i+1}_{LH}, D^{i+1}_{HL}, D^{i+1}_{HH} and S^{i+1}_{LL}, where the former three are high frequency sub-band of vertical, horizontal and diagonal respectively, and the last one is low frequency sub-band. Fig.2 shows a MR brain image and its three levels DWT decomposition. At each level of decomposition the image is filtered and subsampled of a factor 2, which results in four sub-bands. For the first level decomposition, right down HH sub-band is the diagonal detail, right up HL sub-band is the horizontal detail, and left down LH sub-band is the vertical detail. The LL sub-band is iteratively decomposed for further two levels.

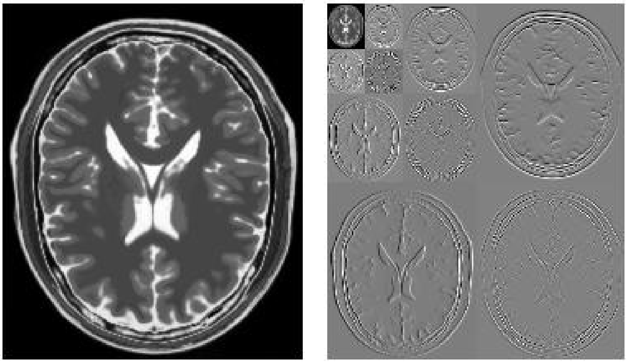


Fig. 2. MR brain image and its DWT decomposition. Left is the original image. Right is the three levels DWT decomposition.

2.2 DWT-Based Image Registration

Due to the inherent multiresolution characteristic in DWT, the coarse-to-fine strategy can be used to speed up image registration. Firstly, the image is decomposed into I levels, I depends on the image size. Level I represents the coarsest level. The registration process starts from the coarsest resolution with the low frequency sub-band S_{LL}^{I}. Then the registration results of S_{LL}^{i} $(i \in [1, I])$ is used as initial position at level $i-1$. So the estimates of the correspondence are gradually improved while going up to the finer resolutions. At every level, the search space and computational time are considerably decreased. The registration process will terminate when the similarity measures are optimized at the highest resolution level.

3 Similarity Measures

The following section outlines six similarity measures used in our experiments. Each similarity measure is used under 2-D rigid transformation. These measures are Partitioned Intensity Uniformity (PIU), Correlation Ratio (CR), Joint Entropy (JE), Mutual Information (MI), Normalized Mutual Information (NMI), and Energy(EN).

3.1 Partitioned Intensity Uniformity (PIU)

PIU was the first widely used multi-modality similarity measure based on pixel intensity, proposed by Woods *et al.* for MR-PET registration [8]. Let Ω denote the overlapping area of image A and B, N the total pixels number in image, and n_a, n_b

denotes the number of pixels with intensity value a and b in Ω, respectively. The definition of PIU can be written as:

$$PIU = \sum_a \frac{n_A}{N} \frac{\sigma_B(a)}{\mu_B(a)} + \sum_b \frac{n_B}{N} \frac{\sigma_A(b)}{\mu_A(b)} \tag{2}$$

where

$$\mu_B(a) = \frac{1}{n_a} \sum_{\Omega_a} B(x_A),\ \mu_A(b) = \frac{1}{n_b} \sum_{\Omega_b} A(x_B) \tag{3}$$

$$\sigma_B(a) = \frac{1}{n_a} \sum_{\Omega_a} \left(B(x_A) - \mu_B(a)\right)^2, \sigma_A(b) = \frac{1}{n_b} \sum_{\Omega_b} \left(A(x_B) - \mu_A(b)\right)^2 \tag{4}$$

$\sum_{\Omega_a} B(x_A)$ is the sum of intensity in image B, which corresponding counterparts in image A have the same intensity $x_A = a$. $\sum_{\Omega_b} A(x_B)$ is determined similarly.

3.2 Correlation Ratio (CR)

Correlation ratio is an algorithm based on standard statistics, proposed by A Roche et al [9]. Let Ω denote the overlapping area of image A and B, and N is the total number of pixels it contains. We consider the iso-sets of $A, \Omega_i = \{\omega \in \Omega, A(\omega) = i\}$, and their cardinals $N_i = Card(\Omega_i)$. The total and conditional moments (mean and variance) of B can be written as:

$$\sigma^2 = \frac{1}{N} \sum_{\omega \in \Omega} B(\omega)^2 - m^2 \quad m = \frac{1}{N} \sum_{\omega \in \Omega} B(\omega) \tag{5}$$

$$\sigma_i^2 = \frac{1}{N_i} \sum_{\omega \in \Omega_i} B(\omega)^2 - m_i^2 \quad m_i = \frac{1}{N_i} \sum_{\omega \in \Omega_i} B(\omega) \tag{6}$$

Then the Correlation ratio (CR) can be defined as:

$$CR = 1 - \frac{1}{N\sigma^2} \sum_i N_i \sigma_i^2 \tag{7}$$

3.3 Entropy-Based Measures and Energy

Assuming a and b is the intensity of image A and B respectively. The joint probability $p(a,b)$ can be obtained by normalizing the joint histogram $h(a,b)$ of the image pair as:

$$p(a,b) = \frac{h(a,b)}{\sum_{a,b} h(a,b)} \tag{8}$$

From the joint probability function, the two marginal probability functions can be obtained directly as:

$$p(a) = \sum_{b} p(a,b)\,,\ \ p(b) = \sum_{a} p(a,b) \tag{9}$$

Three well-known entropy-based measure, Joint Entropy (JE), Mutual Information (MI) and Normalized Mutual Information (NMI) [10-11], can be define as:

$$JE = \sum_{a,b} p(a,b)\log p(a,b) \tag{10}$$

$$MI = \sum_{a,b} p(a,b)\log\frac{p(a,b)}{p(a)p(b)} \tag{11}$$

$$NMI = \frac{2\sum_{a,b} p(a,b)\log\frac{p(a,b)}{p(a)p(b)}}{\sum_{a} p(a) + \sum_{b} p(b)} \tag{12}$$

A simple measure similar to Eq.12 is Energy (EN), which is defined as follows:

$$EN = \sum_{a,b} p(a,b)^2 \tag{13}$$

4 Experimental Results

In this paper, the similarity measures were compared using the dataset with 2D rigid transformation. The test data sets were obtained from BrainWeb, a website providing simulated MR brain images (http://www.bic.mni.mcgill.ca/brainweb/). All the images have been precisely registered beforehand as “golden standard”. Three data sets were used in the experiments: rectified images, images with 9% white Gaussian noise and inhomogeneity images with 40% intensity non-uniformity. One sample slice of the dataset is shown in Fig.3.

T1-weighted images were selected as the reference image, T2-weighted and PD images as floating images. The Haar filter was selected as the wavelet basis function since it can provide easier computation than other basis, and the sub-band S_{LL} was used to perform registration because it preserves most of the significant information of the original image [12]. Because the MR image size is 217×181 , so the wavelet decomposition level is set to 3. Partial volume interpolation and Powell method was adopted as the interpolation and optimization strategy [13]. The estimated parameters of rigid transformation with various similarity measures are closely related to the stop criteria of the optimization process. To make fair comparisons, the stop criterion is the maximal value of the similarity measures.

Reference [14] has evaluated the effect of different grey levels on rigid registration performance of MR and CT images. Considering the accuracy of registration result, computation time cost on whole registration procedure and the perceptibility of human eye, they concluded that rescaling the intensity values of the original images into [0,63] is an excellent tradeoff. But our experimental results with 64 grey levels demonstrated the entropy-based metrics could not achieve subpixel precision for approximation sub-band images. So we rescaled the grey levels of original and decomposed sub-band images into 32 grey levels.

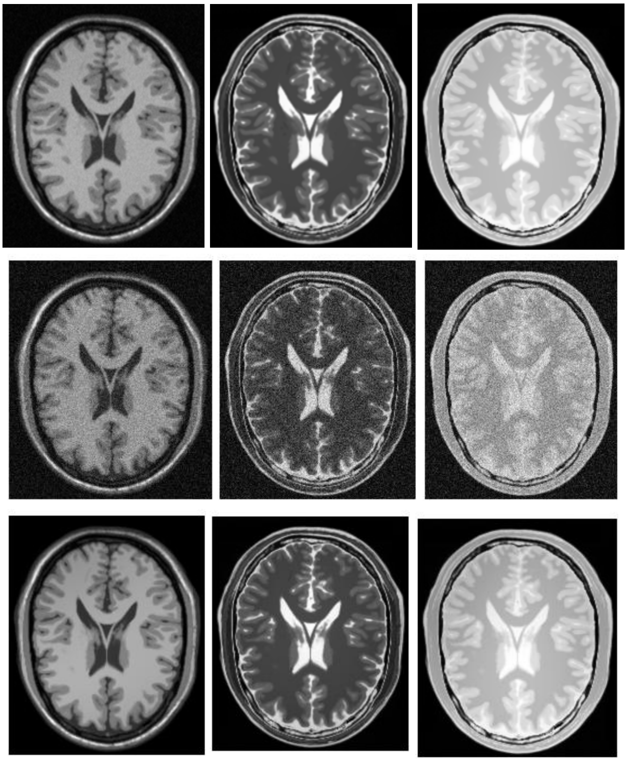


Fig. 3. The 91st slice of the database. The left column is T1-weighted images, the second column is T2-weighted images, and the right column is PD-weighted images. The upper row is rectified images, the second row is images with 9% white Gaussian noise, and the bottom row is images with 40% intensity non-uniformity.

We used 30 randomly selected slices on a specified grid transform parameter, where the X-translation is 4 mm, Y-translation is 4 mm, and rotation is 4 degree. The experimental results are listed in Table 1 to Table 3, where, *mean* and *ratio,* denote the average error of subpixel precision results, and the ratio of achieving subpixel precision. Each component of *mean* represents X-translation mean error, Y-translation mean error, orientation mean error in order. Furthermore, "——" denotes that there is no subpixel precision result.

From Table 1 to Table 3, we can obtain following observations.

1) Performances on the original images

MI and NMI measures can get 100% successful rate in sub-pixel registration and give the highest accuracy fore both T1-T2 and T1-PD to all the three case, namely, rectified, noisy and inhomogeneity. And their performances are almost the same. For rectified case, PIU, CR and JE perform similarly for T1-T2 and T1-PD. However, EN is worst one. For noisy T1-T2 images, JE is worse than MI, NMI and much better than other measures. For noisy T1-PD images, the accuracy of PIU, CR and JE follow closely the MI and NMI. For inhomogeneity case, PIU and EN are the worst ones.

2) Performance on the level 1 approximation sub-band image

For rectified case, MI and NMI are the best ones for both image sets. For noisy case, PIU, MI and NMI perform best among the measure for T1-T2 and PIU is the best one for T1-PD, which followed by MI and NMI. This observation shows that PIU has the best robustness for Gaussian noise. CR and EN have the worst results for noisy case. For inhomogeneity case, MI and NMI are the best measures for T1-T2 and CR is the best one for T1-PD, which followed by MI and NMI. PIU, CR and EN performs worst for T1-T2, but for T1-PD, PIU, JE and EN are the worst ones.

3) Performance on the level 2 approximation sub-band image

For rectified and noisy case, PIU is the best one for both T1-T2 and T1-PD. For inhomogeneity case, CR performs the best for both T1-T2 and T1-PD and followed by PIU. However, the entropy-based measures, JE, MI and NMI, can not get the sub-pixel registration results for all the three cases.

Table 1. Registration results of rectified MR images

	Similarity Measures	Original		Level 1		Level 2	
		mean	*ratio*	*mean*	*ratio*	*mean*	*ratio*
T1 to T2	PIU	0.4,0.1,0.2	93%	0.2,0.0,0.2	43%	0.6,0.1,0.5	10%
	CR	0.5,0.1,0.4	93%	0.3,0.0,0.2	23%	0.6,0.1,0.2	7%
	JE	0.2,0.3,0.0	97%	0.2,0.3,0.0	47%	——	0%
	MI	0.2,0.3,0.0	100%	0.2,0.3,0.0	90%	——	0%
	NMI	0.3,0.1,0.2	100%	0.2,0.3,0.0	90%	——	0%
	EN	0.2,0.4,0.4	67%	0.2,0.6,0.1	33%	0.2,0.4,0.5	10%
T1 to PD	PIU	0.4,0.1,0.1	93%	0.2,0.1,0.3	50%	0.2,0.3,0.4	20%
	CR	0.2,0.2,0.2	93%	0.3,0.2,0.2	73%	0.1,0.2,0.3	7%
	JE	0.2,0.3,0.1	93%	0.2,0.3,0.1	77%	——	0%
	MI	0.2,0.3,0.0	100%	0.2,0.3,0.1	83%	——	0%
	NMI	0.2,0.3,0.0	100%	0.2,0.3,0.1	83%	——	0%
	EN	0.2,0.5,0.5	40%	0.2,0.6,0.2	30%	0.3,0.4,0.5	10%

Table 2. Registration results of noisy MR images

	Similarity Measures	Original		Level 1		Level 2	
		mean	*ratio*	*mean*	*ratio*	*Mean*	*ratio*
T1 to T2	PIU	0.4,0.4,0.3	17%	0.3,0.1,0.2	83%	0.4,0.1,0.5	13%
	CR	0.4,0.5,0.1	40%	0.3,0.3,0.2	37%	——	0%
	JE	0.2,0.3,0.1	93%	0.2,0.3,0.0	77%	——	0%
	MI	0.2,0.3,0.1	100%	0.2,0.3,0.1	83%	——	0%
	NMI	0.3,0.3,0.2	100%	0.2,0.3,0.1	83%	——	0%
	EN	0.2,0.3,0.1	83%	0.3,0.3,0.2	37%	0.7,0.1,0.9	3%
T1 to PD	PIU	0.4,0.3,0.1	93%	0.3,0.1,0.2	87%	0.1,0.1,0.3	33%
	CR	0.4,0.4,0.3	93%	0.4,0.3,0.3	47%	0.9,0.2,0.5	3%
	JE	0.2,0.3,0.1	93%	0.2,0.3,0.1	77%	——	0%
	MI	0.2,0.3,0.1	97%	0.2,0.3,0.1	83%	——	0%
	NMI	0.3,0.3,0.1	97%	0.2,0.3,0.1	83%	——	0%
	EN	0.2,0.4,0.2	87%	0.2,0.3,0.2	43%	——	0%

Table 3. Registration results of inhomogeneity MR images

	Similarity Measures	Original		Level 1		Level 2	
		mean	*ratio*	*mean*	*ratio*	*mean*	*ratio*
T1 to T2	PIU	0.3,0.2,0.3	30%	0.2,0.1,0.2	17%	0.0,0.1,0.7	3%
	CR	0.1,0.1,0.2	97%	0.2,0.1,0.7	37%	0.2,0.1,0.3	10%
	JE	0.2,0.3,0.1	97%	0.2,0.3,0.1	30%	——	0%
	MI	0.2,0.3,0.0	100%	0.2,0.3,0.0	83%	——	0%
	NMI	0.2,0.3,0.0	100%	0.2,0.3,0.0	83%	——	0%
	EN	0.1,0.4,0.4	70%	0.1,0.5,0.2	37%	0.2,0.5,0.9	7%
T1 to PD	PIU	0.3,0.5,0.3	40%	0.5,0.4,0.3	30%	0.1,0.2,0.1	7%
	CR	0.2,0.2,0.3	97%	0.3,0.2,0.2	87%	0.1,0.1,0.4	10%
	JE	0.2,0.3,0.1	97%	0.2,0.1,0.1	53%	——	0%
	MI	0.1,0.3,0.0	100%	0.2,0.3,0.1	73%	——	0%
	NMI	0.2,0.3,0.0	97%	0.2,0.3,0.1	73%	——	0%
	EN	0.1,0.5,0.6	46%	0.1,0.6,0.2	17%	——	0%

So the following useful guidance for multiresolution MR image registration can be drawn.

1) For the original images with highest resolution, MI and NMI are the first choice. They have the distinctive accuracy and robustness even for noise and inhomogeneity cases.

2) For middle resolution, MI and NMI are the best ones for rectified and inhomogeneity images. However, PIU performs the best for noise images.
3) For low-resolution cases, PIU are the best choice for rectified and noisy images, but CR is the best one for inhomogeneity images.

5 Summary

In this paper, we evaluated the performances of six similarity measures at 3 levels of wavelet pyramid with MR images. The experimental results give some guidance to the selection of appropriate similarity measures for MR image registration in a multiresolution wavelet framework. Future work will include comparing these measures in a large number of datasets with non-rigid transformation.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China (No.60402024), Program for New Century Excellent Talents in University, and the Excellent Youth Foundation of Hunan Province.

References

1. Maintz, J.B., Viergever, M.A.: A Survey of Medical Image Registration. Medical Image Analysis 1 (1998) 1-36
2. Zitova´, B., Flusser, J.: Image Registration Methods: A Survey. Image and Vision Computing,21 (2003) 977-1000
3. Hill, D, Batchelor, P., Holden, M., et al.: Medical Image Registration. Physics in Medicine and Biology 46 (2001) 1-45
4. Pluim, J., Maintz, J., Viergever, M.: Mutual Information Matching in Multiresolution Contexts. Image and Vision Computing 19 (2001) 45-52
5. Thevenaz, P., Ruttimann, U., Unser, M.: A Pyramid Approach to Subpixel Registration Based on Intensity. IEEE Transaction on Image Processing 7(1998) 27-41
6. Cole-Rhodes, A., Johnson, K., LeMoigne, J., et al.: Multiresolution Registration of Remote Sensing Imagery by Optimization of Mutual Information Using a Stochatistic Gradient. IEEE Transaction on Image Processing 12(2003) 1495- 1511
7. Moigne, J.L., Campbell, W.J., Cromp, R. F.: An Automated Parallel Image Registration Technique Based on the Correlation of Wavelet Features. IEEE Transaction on Geoscience &Remote Sensing 40(2002) 1849–1864
8. Woods, R.P., Mazziotta, J.C., Cherry, S.R.: MRI-PET Registration with Automated Algorithm. Journal of Computer Assisted Tomography 17(1993) 536-546
9. Roche, A., Malandain, G., Pennec, X., et al.: The Correlation Ratio as a New Similarity Measure for Multimodal Image Registration. Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention (1998) 1115-1124
10. Maes, F., Collignon, A., Vandermeulen, D., et al.: Multimodality Image Registration by Maximization of Mutual Information. IEEE Transactions on Medical Imaging 16(1997) 187–198

11. Studholme, C., Hill, D.L.G., Hawkes, D.J.: An Overlap Invariant Entropy Measure of 3D Medical Image Alignment. Pattern Recognition 32(1999) 71-86.
12. Wu, J., Chung, A.: Multimodal Brain Image Registration Based on Wavelet Transform using SAD and MI. The Second International Workshop on Medical Imaging and Augmented Reality (2004) 270-277
13. Maes, F., Collignon, A., Vandermeulen, D., et al.: Multimodality Image Registration by Maximization of Mutual Information. IEEE Transaction on Medical Imaging 16 (1997) 187-198
14. Gao, Z., Lin, J., Xu, B.: The Affection of Grey Levels on Mutual Information Based Medical Image Registration. Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2003) 1575- 1579
15. Škerl, D., Likar, B., Pernuš, F.: A Protocol for Evaluation of Similarity Measures for Rigid Registration. IEEE Transaction on Medical Imaging 25 (2006) 779-791

Finding the Shortest Path Between Two Points in a Simple Polygon by Applying a Rubberband Algorithm

Fajie Li and Reinhard Klette

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

Abstract. Let p and q be two points in a simple polygon Π. An open problem in computational geometry asks to devise a simple linear-time algorithm for computing a shortest path between p and q, which is contained in Π, such that the algorithm does not depend on a (complicated) linear-time triangulation algorithm. This report provides a contribution to the solution of this problem by applying the rubberband algorithm. The obtained solution has $\mathcal{O}(n\ log\ n)$ time complexity (where the super-linear time complexity is only due to preprocessing, i.e. for the calculation of critical edges) and is, altogether, considerably simpler than the triangulation algorithm. It has applications in 2D pattern recognition, picture analysis, robotics, and so forth.

Keywords: digital geometry, computational geometry, rubberband algorithm, simple polygon, Euclidean shortest path.

1 Introduction

So far, methods for computing the Euclidean shortest path (ESP) between two points in a simple polygon, as in [3,4,5,11], all rely on starting with a rather complicated, but linear-time triangulation [2] of a simple polygon. In this paper, we apply a version of a rubberband algorithm to devise a simple $\mathcal{O}(n\ log\ n)$ algorithm for computing a shortest path between p and q, which is contained in Π, where n is the number of vertices of Π. Our algorithm starts with a special (say, horizontal) trapezoidal segmentation of the polygon, which is computationally not very different to a triangulation, and thus our segmentation can also be seen as a possible contribution to simplify the triangulation procedure.

The original rubberband algorithm was published in [1] and [6], aiming at an (approximative) calculation of a minimum-length polygonal path (MLP) contained and complete in a simple cube-curve (subsequent grid cubes of this curve are face-adjacent) in 3D Euclidean space. Three cubes of such a curve form a *corner* if all three are incident with one grid edge (called a *critical edge* of the cube-curve). MLP vertices are always located on critical edges [1].

The correctness and actual time-complexity of this original rubberband algorithm remained an open problem for some time. [8] proved that this algorithm is

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 280–291, 2006.

always correct for some special cases of simple cube-curves. The algorithm was then slightly corrected in [9], which also allowed to prove that its running time is always linear for a special class of input curves. Finally, [10] presented two provably correct and linear-time *edge-based* or *face based* rubberband algorithms for the general case (i.e., arbitrary simple cube-curves). The rubberband algorithm, as applied in this paper, is a simplified version of the edge-based rubberband algorithm, and it is presented in Section 3.1.

There is a general option provided by the studied rubberband algorithms: the basic approach for minimizing a path does not depend upon the specific geometric shape of grid cubes; it can be applied to a wide variety of 2D or 3D path minimization problems where segmentations into convex subsets are appropriate. We illustrate this in this brief note for one 2D example only (and will do for others in forthcoming publications).

In the rest of this paper, Section 2 provides necessary definitions and theorems. Section 3 presents not only our algorithm but also examples and analysis of time complexity. Section 4 concludes the paper.

2 Basics

We denote by $\Pi = \langle v_1, v_2, \ldots, v_n \rangle$ a simple polygon (i.e., a compact polygonal region) in the 2D Euclidean plane (which is equipped with an xy Cartesian coordinate system). $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices of Π, and $\vartheta\Pi = \cup_{i=1}^{n}\{v_i v_{i+1}\}$ (mod n) is a simple polygonal curve specifying edges forming the frontier of Π.

For $p \in \mathbb{R}^2$, let p_x be the x-coordinate of p. Let $s = p_1p_2$, with $p_1, p_2 \in \vartheta\Pi$ and $p_{1_x} \leq p_{2_x}$. Furthermore, assume that s is parallel to the x-axis, $s \subset \Pi$, and there is no $v \in V \backslash \{p_1, p_2\}$ such that v is between p_1 and p_2.

Definition 1. *If $p_1 \in V$ ($p_2 \in V$) then we say that s is the* right *(*left*)* critical segment *(of Π) with respect to p_1 (p_2). If $p_1 = p_2$ then we say that s is* degenerate*. A* critical segment *is either a left or a right critical segment.*

See Figure 1 and Table 1 for examples. – For a critical segment s of Π, let s_y be the y-coordinate of points on s. For a given y, let $\{s_1, s_2, \ldots, s_m\}$ be the

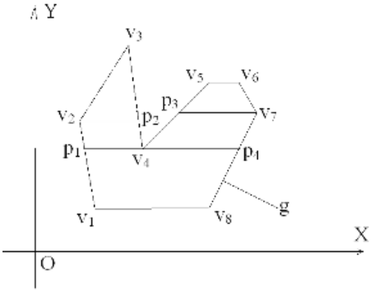


Fig. 1. Critical segments of a simple polygon (see also Table 1)

Table 1. All critical segments of vertices of the simple polygon of Figure 1

Vertex	Left critical segment	Right critical segment
v_1	degenerate	v_1v_8
v_2	degenerate	v_2p_2
v_3	degenerate	degenerate
v_4	p_1v_4	v_4p_4
v_5	degenerate	v_5v_6
v_6	v_5v_6	degenerate
v_7	p_3v_7	degenerate
v_8	v_1v_8	degenerate

maximal set of critical segments of Π such that $s_{i_y} = s_{i+1_y}$ and $s_i \cap s_{i+1} \neq \phi$, where i = 1, 2, ..., m - 1.

Definition 2. *The segment* $\cup_{i=1}^{m}\{s_i\}$ *is a* maximal critical segment *of* Π*. If* $m > 1$ *(*m *= 1) then we say that the segment* $\cup_{i=1}^{m}\{s_i\}$ *is a* non-trivial *(*trivial*)* maximal critical segment *of* Π*.*

In Figure 1, p_1p_4 is the only non-trivial maximal critical segment of the shown simple polygon.

Definition 3. *Two maximal critical segments* s_1 *and* s_2 *are called* adjacent *iff there is no maximal critical segment* s_3 *such that* $s_{1_y} < s_{3_y} < s_{2_y}$ *or* $s_{2_y} < s_{3_y} < s_{1_y}$*.*

In Figure 1, the trivial maximal critical segment v_5v_6 is adjacent to the trivial maximal critical segment p_3v_7, but not adjacent to the non-trivial maximal critical segment p_1p_4.

Let $\{s_1, s_2, \ldots, s_k\}$ be the set of maximal critical segments of Π. Construct a weighted tree T as follows: Let $T = [V, E]$, where $V = \{u_1, u_2, \ldots, u_k\}$, $E = \{u_iu_j : s_i$ and s_j are adjacent $\}$, and each $e \in E$ has a weight equal to 1.

Definition 4. *We say that* T *is a* 1-tree *of* Π *(with respect to the given Cartesian coordinate system).*

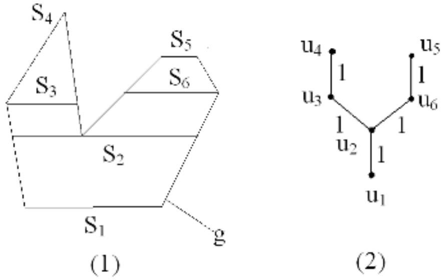


Fig. 2. Left: simple polygon with six maximal critical segments. Right: its 1-tree.

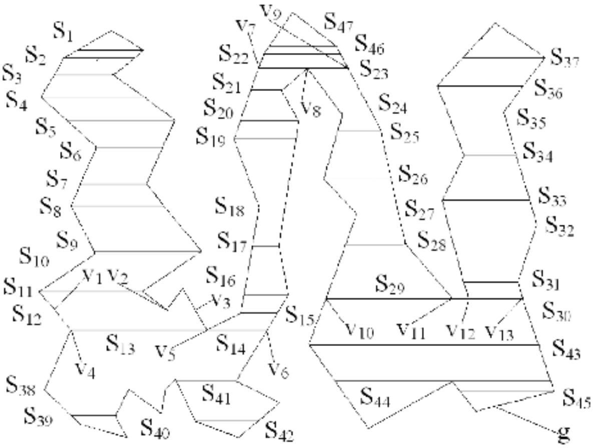


Fig. 3. A “non-trivial” simple polygon Π with its critical segments

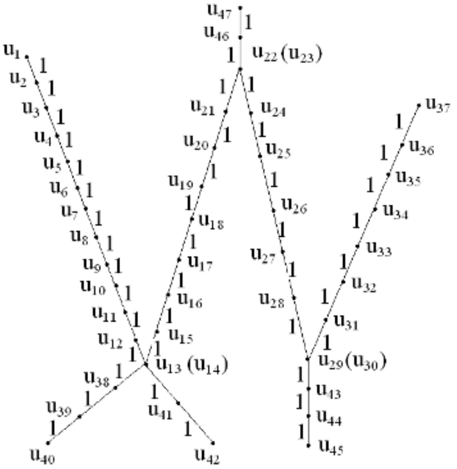


Fig. 4. 1-tree of the simple polygon shown in Figure 3

Figure 2 shows a 1-tree of a simple polygon, and Figure 4 that of the “non-trivial” simple polygon which is shown in Figure 3. – Let $S = \{s_1, s_2, \ldots, s_k\}$ be a subset of the set of all maximal critical segments of Π.

Definition 5. *S is called a* sequence *of maximal critical segments of Π iff, for each $i \in \{1, 2, \ldots, k-1\}$, s_i is adjacent to s_{i+1}.*

If, for each $i \in \{1, 2, \ldots, k-1\}$, $s_{iy} < s_{i+1_y}$ ($s_{iy} > s_{i+1_y}$) then S is called an *increasing (decreasing)* sequence of maximal critical segments of Π. S is called a *monotonous* sequence of maximal critical segments of Π iff it is either an increasing or a decreasing sequence of maximal critical segments of it. Finally, S is called an *alternate* monotonous sequence of maximal critical segments of Π

iff it is a sequence of maximal critical segments of Π, and it is the union of a finite number of monotonous sequences of maximal critical segments of Π.

In Figure 2 (left), $\{s_1, s_2, s_3, s_4\}$ is an increasing sequence of maximal critical segments of Π while $\{s_5, s_6, s_2, s_1\}$ is a decreasing sequence of maximal critical segments. $\{s_3, s_2, s_6\}$ is an alternate monotonous sequence of maximal critical segments of Π.

Let $S = \{s_1, s_2, s_3\}$ be a sequence of maximal critical segments of Π with $s_1 \neq s_3$ such that there is no maximal critical segment between s_1 and s_2 or s_3 and s_2, and there exist critical segments s_1^* and s_2^* such that $s_1^* \subset s_2$ and $s_2^* \subset s_2$, and s_1^* and s_1 are two edges of a quadrilateral, as well as s_2^* and s_3. (For example, in Figure 3, $\{s_{28}, v_{10}v_{13}, s_{31}\}$ is such a sequence of maximal critical segments of Π, with $v_{10}v_{11}$, $v_{12}v_{13} \subset v_{10}v_{13}$; segments $v_{10}v_{11}$ and s_{28} are two edges of a quadrilateral, as well as $v_{12}v_{13}$ and s_{31}.) For such a sequence $S = \{s_1, s_2, s_3\}$ we define the following:

Definition 6. *If $s_{1y} < s_{2y}$ and $s_{2y} > s_{3y}$ ($s_{1y} > s_{2y}$ and $s_{2y} < s_{3y}$) then s_2 is called an* upward *(*downward*) maximal critical segment of Π.*

Furthermore, s_2 is also called a *stable* maximal critical segment of Π if it is an upward or downward maximal critical segment of Π; both s_1^* and s_2^* are called the *good* critical segments of s_2 with respect to Π.

In Figure 2, s_2 is the unique downward maximal critical segment of the shown simple polygon. There is no upward maximal critical segment.

Let $p, q \in \mathbb{R}^2$ be two points in the simple polygon Π. Let $\rho(p, q)$ be a shortest path from p to q. Let $S = \{s_1, s_2, \ldots, s_k\}$ be the set of maximal critical segments of Π such that, for each $i \in \{1, 2, \ldots, k\}$, $s_i \cap \rho(p, q) \neq \phi$.

We modify S as follows: if s_i is a stable maximal critical segment, then replace s_i by its good critical segments.

In Figure 3, v_4v_6 is a stable maximal critical segment. It has two good critical segments v_4v_5 and v_5v_6. Segment $v_{10}v_{13}$ is another stable maximal critical segment. It has two good critical segments $v_{10}v_{11}$ and $v_{12}v_{13}$.

Definition 7. *The modified set S is called a* step set *of maximal critical segments of Π with respect to the shortest path $\rho(p, q)$.*

For example, in Figure 1, $\{v_2p_2, p_1v_4, v_4p_4\}$ (obtained by modifying the set $\{v_2p_2, p_1p_4\}$) is a step set of maximal critical segments of the simple polygon Π with respect to the shortest path $\rho(v_3, v_7)$. As another example, in Figure 3,

$$(\cup_{i=1}^{12}\{s_i\}) \cup \{v_4v_6\} \cup (\cup_{i=15}^{21}\{s_i\}) \cup \{v_7v_9\} \cup (\cup_{i=24}^{28}\{s_i\}) \cup \{v_{10}v_{13}\} \cup (\cup_{i=31}^{37}\{s_i\})$$

is a set of maximal critical segments of Π. It can be modified into a step set

$$\begin{aligned}
&(\cup_{i=1}^{12}\{s_i\}) \cup \{s_{13}(= v_4v_5), s_{14}(= v_5v_6)\} \cup (\cup_{i=15}^{21}\{s_i\}) \cup \\
&\{s_{22}(= v_7v_8), s_{23}(= v_8v_9)\} \cup (\cup_{i=24}^{28}\{s_i\}) \cup \\
&\{s_{29}(= v_{10}v_{11}), s_{30}(= v_{12}v_{13})\} \cup (\cup_{i=31}^{37}\{s_i\})
\end{aligned}$$

Let S be a step set of maximal critical segments of Π with respect to the shortest path $\rho(p, q)$, and $s_1 \in S$. Let $d_e(p, q)$ be the Euclidean distance between points p and q.

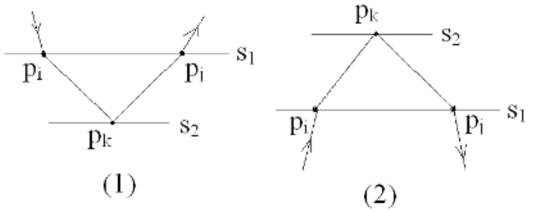


Fig. 5. Illustration for the proof of Lemma 1. Left: *Case 1.* s_1 is a downward maximal critical segment. Right: *Case 2.* s_1 is an upward maximal critical segment.

Lemma 1. *If s_1 is a downward (upward) maximal critical segment of Π and s_2 is a maximal critical segments of Π such that $s_{1_y} > s_{2_y}$ ($s_{1_y} < s_{2_y}$), then $s_2 \notin S$.*

Proof. The proof is by contradiction. Let $\{p_i, p_j\} = s_1 \cap \rho(p, q)$. Suppose that $p_k \in s_2 \cap \rho(p, q)$, then $d_e(p_i, p_j) < d_e(p_i, p_k) + d_e(p_k, p_j)$. Therefore, $\rho(p, q)$ is not a shortest path. (See Figure 5 for an illustration.) □

By Lemma 1 we have the following theorem:

Theorem 1. *If S is a step set of maximal critical segments of Π with respect to the shortest path $\rho(p, q)$ then S is an alternate monotonous sequence of maximal critical segments of Π.*

This theorem and the following, previously known result are important for proving that our ESP Algorithm (described in Section 3.4) requires only linear computation time.

Theorem 2. ([12], Theorem 37) *There is a deterministic linear time and linear space algorithm for the single source shortest path problem for undirected graphs with positive integer weights.*

Let T be a tree and p, q vertices of T.

Corollary 1. *There is a deterministic linear time and linear space algorithm to find the unique path $\rho(p, q) \subset T$.*

3 The Algorithms

A simplified 2D rubberband algorithm will be used in the main algorithm described in Section 3.4.

3.1 A Simplified Rubberband Algorithm

Let $p, q \in \mathbb{R}^2$, $S = \{s_1, s_2, \ldots, s_k\}$ be a set of consecutive, pairwise disjoint nondegenerate critical segments, $P = \{p_1, p_2, \ldots, p_k\}$ such that $p_i \in s_i$, where $i =$ 1, 2, …, k ($k \geq 3$). Let $\rho = \langle p, p_1, p_2, \ldots, p_k, q\rangle$ be a polygonal arc that starts at p, then visits p_1, p_2, …, p_k, and finally ends at q.

Rubberband Algorithm

```
1. Let $\epsilon = 10^{-10}$ (the accuracy).
2. Compute the length $l_1$ of the path $\rho = \langle p, p_1, p_2, \ldots, p_k, q \rangle$.
3. Let $q_1 = p$ and $i = 1$.
4. While $i < k - 1$ do
4.1 Let $q_3 = p_{i+1}$.
4.2 Compute a point $q_2 \in s_i$ such that
        $d_e(q_1, q_2) + d_e(q_3, q_2) = \min\{d_e(q_1, q) + d_e(q_3, q) : q \in s_i\}$
4.3 Update $P$ by replacing $p_i$ by $q_2$.
4.4 Let $q_1 = p_i$ and $i = i + 1$.
5.1 $q_3 = q$.
5.2 Compute $q_2 \in s_k$ such that
        $d_e(q_1, q_2) + d_e(q_3, q_2) = \min\{d_e(q_1, q) + d_e(q_3, q) : q \in s_k\}$
5.3 Update $P$ by replacing $p_k$ by $q_2$.
6. Compute the length $l_2$ of the updated path $\rho = \langle p, p_1, p_2, \ldots, p_k, q \rangle$.
7. Let $\delta = l_1$ - $l_2$.
8. If $\delta > \epsilon$, then let $l_1 = l_2$ and go to Step 3. Otherwise stop.
```

The accuracy parameter in Step 1 can be chosen such that maximum possible numerical accuracy is guaranteed on the given computer.

3.2 Examples

In Figure 6, (upper row, left) shows start and destination points p and q, and three critical segments s_1, s_2 and s_3. (Upper row, middle) shows the initial points p_1, p_2 and p_3 which are the centers of s_1, s_2 and s_3, respectively. (Upper row, right), (lower row, left) and (lower row, middle) show updated points p_1 (by step 4.3), p_2 (by step 4.3) and p_3 (by step 5.3), respectively. (Lower row, right) shows the final path.

In Figure 7, (left) shows the initial path ρ_0, and the updated paths ρ_1 to rho_4 of four iterations of the Rubberband Algorithm. (Right) shows the initial path ρ_0, and the updated paths ρ_1 to ρ_{18} of eighteen iterations of this algorithm.

We can see that different initial points may lead to different numbers of iterations of the algorithm until it terminates (with respect to the chosen accuracy parameter). From Figure 6, we can see that the algorithm only needs two iterations to terminate. The results of the first iteration are shown in (upper row, right), (lower row, left) and (lower row, middle). (Lower row, right) shows the result of the second iteration. Figure 7 (left) shows that the algorithm has to run for at least 4 iterations in this case.

3.3 Time Complexity

Let ϵ be the accuracy, l the true length of the shortest path, l_0 that of initial path, and l_n that of the path after n-iteration. We slightly modify the Rubberband

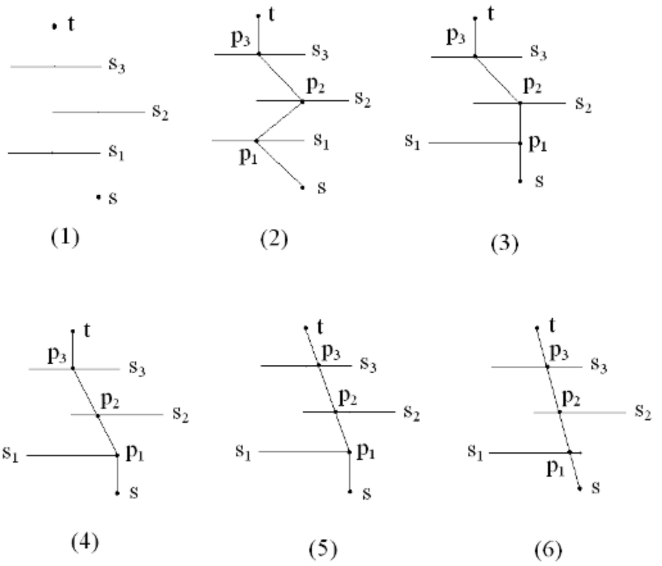


Fig. 6. Illustration for the Rubberband Algorithm when the initial points are selected as centers of critical segments

Algorithm as follows:[1] For each iteration, we update the vertices with odd indices first and then update those with even indices later (i.e., for each iteration, we update the vertices with indices 1, 3, 5, …, then those with indices 2, 4, 6, …), then $\{l_n\}$ is a decreasing sequence with lower bound 0. Let $l_0 - l = ak + b$ and $l_n - l_{n+1} = ck + d$, where a, b, c and d are constants such that a, $c \neq 0$. Then we have

$$\lim_{k\to\infty} \frac{ak+b}{ck+d} = \frac{a}{c}$$

Therefore the algorithm will stop after at most $\lceil a/(c\epsilon)\rceil$ iterations. So the time complexity of the Rubberband Algorithm is $\lceil a/(c\epsilon)\rceil \cdot \mathcal{O}(k) = \mathcal{O}(k)$, where k is the number of the vertices of the path.

3.4 New Algorithm

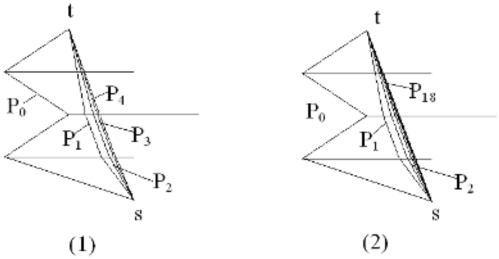
Let p, q be the start and destination point inside of a simple polygon Π, respectively. Let V be the set of vertices of Π. Let E be the set of edges of Π.

Preprocessing Procedure

1. The sorted set $V = \{v_1, v_2, \ldots, v_n\}$ be the set of vertices of Π such that $v_{i_y} \leq v_{i+1_y}$, where i = 1, 2, …, n - 1.

2. For each $v_i \in V$, compute a straight line l_i such that l_i is parallel to x-axis.

[1] This is just for the purpose of time complexity analysis. By experience, the Rubberband Algorithm runs faster without such a modification.



(1) (2)

Fig. 7. Illustration for the Rubberband Algorithm when the initial points are the left end points of critical segments

3. For each $e \in E$,

3.1 let $e = v_i v_j$ and $i < j$, and i, j = 1, 2, ..., n - 1;

3.2 if e is parallel to x-axis, let $S_i = S_i \cup \{p\}$ and $S_j = S_j \cup \{p\}$;

3.3 otherwise, for each $m \in \{i, i+1, \ldots, j-1\}$, let $p = l_m \cap e$ and $S_m = S_m \cup \{p\}$.

4. For each $i \in \{1, 2, \ldots, n\}$, find $v_i \in S_i$.

4.1 Let

$$v_{ileft} = \max\{v_i{}' : v_{i_x}' \leq v_{ix} \wedge v_i{}' \in S_i\}$$

and

$$v_{iright} = \min\{v_i{}' : v_{i_x}' \geq v_{ix} \wedge v_i{}' \in S_i\}$$

(It follows that $v_{ileft} v_i$ and $v_i v_{iright}$ are the left and right critical segments of v_i, respectively.)

5. Partition V into V_1, V_2, ..., V_k such that $V_i = \{v_{i1}, v_{i2}, \ldots, v_{in_i}\}$, $v_{ij_y} = v_{ij+1_y}$, where j = 1, 2, ..., n_i - 1, and $v_{i1_y} < v_{i+11_y}$, where i = 1, 2, ..., k - 1.

6. Merge all left and right critical segments of $v \in V_i$ into a maximal critical segment of Π, denoted by s_v.

7. Output $S = \{s_v : v \in V_i,\ i = 1, 2, \ldots, k\}$.

ESP Algorithm

1. Apply the Preprocessing Procedure to compute the set of maximal critical segments of Π, denoted by S.

2. Construct a 1-tree T.

3. Apply the algorithm of [12] to find the unique path $\rho(p, q) \subset T$.

4. Compute the step set of maximal critical segments of Π with respect to the shortest path $\rho(p, q)$, denoted by S_{step} (see description before Definition 7).

5. Let $P = \{p\}$.

6. For each $s_i \in S_{step}$,

6.1 let v_i be the center point of s_i;

6.2 let $P = P \cup \{v_i\}$;

6.3 let $P = P \cup \{q\}$.

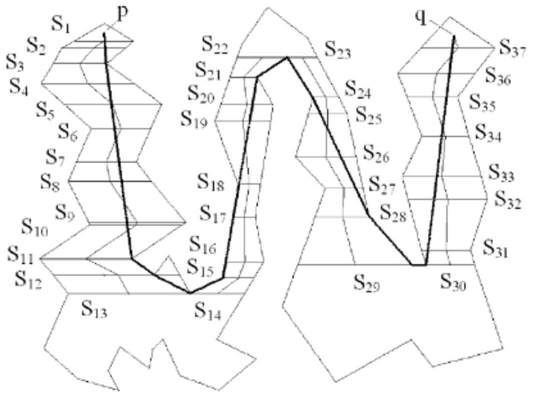


Fig. 8. The shortest path from s to t inside the simple polygon shown in Figure 3

7. Apply the Rubberband Algorithm on S_{step} and P to compute the shortest path $\rho(p, q)$ inside of Π.

8. We finally convert $\rho(p, q)$ into the standard form of the shortest path by deleting all vertices which are not vertices of Π.[2]

Table 2. Vertices (not including p and q) of the shortest path $\rho(p, q)$ obtained by Step 7 in the ESP Algorithm, for the example shown in Figure 3

i	(x_i, y_i)	i	(x_i, y_i)	i	(x_i, y_i)	i	(x_i, y_i)
1	(281.0, 734.0)	11	(342.0, 284.0)	21	(625.0, 659.0)	31	(1010.1, 302.0)
2	(281.9, 719.0)	12	(392.0, 250.0)	22	(693.0, 700.0)	32	(1024.2, 408.0)
3	(284.0, 687.0)	13	(474.0, 212.0)	23	(693.0, 700.0)	33	(1030.4, 454.0)
4	(289.9, 646.0)	14	(474.0, 212.0)	24	(750.0, 617.0)	34	(1041.4, 537.0)
5	(296.1, 603.0)	15	(548.0, 244.0)	25	(767.1, 584.0)	35	(1052.2, 618.0)
6	(303.3, 553.0)	16	(554.3, 278.0)	26	(813.8, 494.0)	36	(1058.7, 667.0)
7	(313.2, 484.0)	17	(571.2, 369.0)	27	(847.5, 429.0)	37	(1065.8, 720.0)
8	(319.0, 444.0)	18	(584.2, 439.0)	28	(877.0, 372.0)		
9	(331.2, 359.0)	19	(608.1, 568.0)	29	(974.0, 271.0)		
10	(331.9, 354.0)	20	(614.4, 602.0)	30	(1006.0, 271.0)		

We show that Step 8 is always correct. First, let p be a point in the set of vertices of the shortest path $\rho(p, q)$ obtained by Step 7. p must be deleted even if it is close to some vertex of Π. To see this, note that each vertex of the polygon is an endpoint of a critical segment. When we update a point on a critical segment, we search for the new position on the whole segment including its two endpoints. Any really "good" endpoint will be selected quickly. This is illustrated by the example output in Tables 2 and 3. For a point on the shortest path, if it is really "good" then it must be exactly a vertex of the polygon, not just "close"

[2] It is well known that each vertex ($\neq p$, q) of the shortest path is a vertex of Π ([7]).

Table 3. Vertices (not including p and q) of the standard form of the shortest path $\rho(p, q)$ obtained by Step 8 in the ESP Algorithm, for the example shown in Figure 3

i	(x_i, y_i)	i	(x_i, y_i)	i	(x_i, y_i)	i	(x_i, y_i)
3	(284, 687)	14	(474, 212)	23	(693, 700)	30	(1006, 271)
11	(342, 284)	15	(548, 244)	24	(750, 617)		
12	(392, 250)	21	(625, 659)	28	(877, 372)		
13	(474, 212)	22	(693, 700)	29	(974, 271)		

to some vertex. Secondly, let p_j, p_{j+1}, p_{j+2} be three consecutive points in the set of vertices of the shortest path $\rho(p, q)$ obtained by Step 7. If p_{j+1} must be deleted, then $p_j p_{j+2}$ must be contained inside Π. This is because, if p_{j+1} is not a vertex of the polygon, then p_j, p_{j+1} and p_{j+2} must be colinear. Otherwise, there is a point p'_{j+1}in a sufficiently small neighborhood of p_{j+1} such that p'_{j+1} is contained in the polygon and

$$d_e(p_j, p'_{j+1}) + d_e(p'_{j+1}, p_{j+2}) < d_e(p_j, p_{j+1}) + d_e(p_{j+1}, p_{j+2})$$

(This contradicts that $p_j p_{j+1} p_{j+2}$ is the shortest subpath of the shortest path). Since p_j, p_{j+1}, p_{j+2} are colinear, so $p_j p_{j+2}$ is contained in the polygon. Therefore, if a point in the set of vertices of $\rho(p, q)$ is not a vertex of the polygon, then it must be redundant and must be deleted.

Figure 8 shows the initial path and the shortest path (inside the simple polygon shown in Figure 3) computed by the ESP Algorithm. Table 4 illustrates (for the same input example) the relationship between number of iterations and length differences, for the last two updated paths. Initial points are the center points of the segments.

Table 4. Number of iterations (I) and resulting δ, for the example shown in Figure 3

I	δ	I	δ	I	δ	I	δ
1	90.8685	6	0.3787	11	0.0170	16	0.0009
2	34.1894	7	0.2328	12	0.0078	17	0.0006
3	6.9828	8	0.1547	13	0.0043	18	0.0004
4	2.3697	9	0.0992	14	0.0025	19	0.0003
5	0.8061	10	0.0384	15	0.0015	20	0.0002

3.5 Time Complexity of the ESP Algorithm

The main step of the Preprocessing Procedure is Step 3.3. Since each vertex can only have a left critical segment and a right critical segment, the total total number of intersections is not more than $2n$, where n is the number of edges of Π. We assume to apply sorting in Step 1 of the Preprocessing Procedure, so the complexity of this procedure equals $\mathcal{O}(n\ log\ n)$. Therefore, the total time complexity of the ESP Algorithm is also $\mathcal{O}(n\ log\ n)$.

4 Conclusion

We have described a simple $\mathcal{O}(n\ log\ n)$ algorithm for computing a shortest path between p and q, which is contained in a simple polygon Π, where n is the number of vertices of Π. The maximum number of iterations is a constant defined by the selected accuracy parameter. The algorithm is easy to implement.

Obviously, our method can also be generalized to deal with special cases of 3D Euclidean shortest paths. However, this short note is only an initial illustration how versions of a rubberband algorithm apply to shortest path problems.

Acknowledgements. The authors thank Joe Mitchell and David Eppstein for critical comments about an earlier draft of this report.

References

1. T. Bülow and R. Klette. Digital curves in 3D space and a linear-time length estimation algorithm. *IEEE Trans. Pattern Analysis Machine Intelligence*, **24**:962–970, 2002.
2. B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Computational Geometry*, **6**:485–524, 1991.
3. L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, **2**:209–233, 1987.
4. L. Guibas, J. Hershberger Optimal shortest path queries in a simple polygon. *J. Computer System Sciences* **39**:126–152, 1989.
5. J. Hershberger. A new data structure for shortest path queries in a simple polygon. *Information Processing Letters*, **38**:231-235, 1991.
6. R. Klette and A. Rosenfeld. *Digital Geometry: Geometric Methods for Digital Picture Analysis.* Morgan Kaufmann, San Francisco, 2004.
7. D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks* **14**:393–410, 1984.
8. F. Li and R. Klette. Minimum-Length Polygons of First-Class Simple Cube-Curve. In Proc. *Computer Analysis Images Patterns*, LNCS 3691, pages 321–329, Springer, Berlin, 2005.
9. F. Li and R. Klette. Analysis of the rubberband algorithm. Technical Report CITR-TR-175, Computer Science Department, The University of Auckland, Auckland, New Zealand, 2006 (www.citr.auckland.ac.nz).
10. F. Li and R. Klette. Shortest paths in a cuboidal world. In Proc. *Int. Workshop Combinatorial Image Analysis*, LNCS 4040, pages 415–429, Springer, Berlin, 2006.
11. J. S. B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack, J. Urrutia, eds, *Handbook of Computational Geometry*, pages 633–701. Elsevier Science Publishers, 2000.
12. M. Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *J. ACM*, **3**:362–394, 1999.

Facial Expressions Recognition in a Single Static as well as Dynamic Facial Images Using Tracking and Probabilistic Neural Networks

Hadi Seyedarabi[1], Won-Sook Lee[2], Ali Aghagolzadeh[1],
and Sohrab Khanmohammadi[1]

[1] Faculty of Electrical and Computer Engineering
University of Tabriz,Tabriz, Iran
[2] School of Information Technology and Engineering (SITE)
Faculty of Engineering, University of Ottawa, Canada
{seyedarabi, aghagol, khan}@tabrizu.ac.ir, wslee@uottawa.ca

Abstract. An efficient, global and local image-processing based extraction and tracking of intransient facial features and automatic recognition of facial expressions from both static and dynamic 2D image/video sequences is presented. Expression classification is based on Facial Action Coding System (FACS) a lower and upper face action units (AUs), and discrimination is performed using Probabilistic Neural Networks (PNN) and a Rule-Based system. For the upper face detection and tracking, we use systems based on a novel two-step active contour tracking system while for the upper face, cross-correlation based tracking system is used to detect and track of Facial Feature Points (FFPs). Extracted FFPs are used to extract some geometric features to form a feature vector which is used to classify input image or image sequences into AUs and basic emotions. Experimental results show robust detection and tracking and reasonable classification where an average recognition rate is 96.11% for six basic emotions in facial image sequences and 94% for five basic emotions in static face images.

Keywords: Active contours, Action Units, Facial Expressions, Probabilistic Neural Networks.

1 Introduction

Automated facial expression analysis using computer vision work could bring facial expressions into man-machine interaction.

Most computer-vision based approaches to facial expression analysis attempt to recognize only prototypic emotions. These prototypic emotional seem to be universal across human ethnicities and cultures and comprise happiness, sadness, fear, disgust, surprise, and anger. In everyday life, however, such prototypic expressions occur relatively infrequently. Instead, emotion is communicated by changes in one or two discrete features.

In order to make the recognition procedure more standardized, a set of facial muscle movements (known as Action Units) that produce each facial expression, was

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 292 – 304, 2006.

created by psychologists as Facial Action Coding System (FACS) [1]. Table 1 shows AUs used in this work that occur in the lower and upper face and are more important in describing facial expressions.

In recent years, there has been extensive research on facial expression analysis and recognition.

Pantic and Rothkrantz [2] proposed an expert system for automatic analysis of facial expressions from static face images. Their system consists of two major parts, the first one forms a frame work for hybrid facial feature detection and the second part of the system converts low level face geometry into high level facial actions.

Table 1. Some of FACS AUs used in this work

AU (Upper Face)	**FACS description**	**AU (Lower Face)**	**FACS description**
1	Raised inner brows	12	Mouth corners pulled up
2	Raised outer brows	15	Mouth corners pulled downwards
4	Lowered brows	17	Raised chin
5	Raised upper lid	20	Mouth stretched
6	Raised cheek	23	Lips tightened
7	Raised lower lid	24	Lip pressed
9	Wrinkled nose	25	Lips parted
-	-	26	Jaw dropped
-	-	27	Mouth stretched

Lien et al. [3] developed a facial expressions recognition system that was sensitive to subtle changes in the face. The extracted feature information, using a wavelet motion model, was fed to discrimination classifiers or hidden markov models that classified it into FACS action units. The system was tested on image sequences from 100 subjects of varied ethnicity. Average recognition accuracy for 15 AUs in the brow, eye and mouth regions was 81-91%.

Valstar et al. [4] used temporal templates which were 2D images, constructed from image sequences and showed where and when motion in the image sequences has occurred. A K-Nearest Neighbor algorithm and a rule-based system performed the recognition of 15 AUs occurring alone or in combination in an input face image sequences. Their proposed method achieved an average recognition rate of 76.2% on the Cohn-Kanade face image database.

Bartlett et al. [5] used Gabor filters using AdaBoost for feature selection technique followed by classification with Support Vector Machines. The system classified 17 AUs with a mean accuracy of 94.8%. The system was trained and tested on Cohn-Kanade face image database.

In this paper we develop an automatic facial expressions analysis and classification systems. Estimated positions of lips, eyes and eyebrows are determined by using a knowledge based system.

In the first frame of image sequences, 25 Facial Feature Points (FFPs) are automatically detected, using active contours for the lower face and gray level projection method for the upper face. A hybrid tracking system is used to track these FFPs in subsequent frames. An enhanced version of the conventional active contour tracking system is used for lip tracking and a cross-correlation based tracking system is used to track FFPs around eye, eyebrow and nose. Some geometric features are extracted, based on the position of FFPs in the first and the last frames. This features form a feature vector which is used for classifying of input image sequences into 16 AUs using Probabilistic Neural Networks (PNN). A rule-based decision making system is applied to AUs to classify input images into basic emotions.

Proposed features and feature extraction method can also be applied to static images (except features for wrinkled nose). Last frame in image sequences which represents peak of facial expressions is used to train and test of static images recognition system. A local reference parameter is used to normalize extracted geometric features.

Most of the facial expression recognition systems use manually located FFPs in the first frame [3-5]. Our proposed system uses automatically detecting and tracking of feature points. Proposed hybrid tracking system shows robust tracking results both in upper and lower face, which only needs the rough estimated position of eye, eyebrow and mouth.

The system is trained and tested on 180 image sequences, consisting six basic facial expressions on Cohn-Kanade face image database [6].

2 Initial Position of Facial Features

Among the facial features, eye, eyebrow and mouth have important role in expressing facial emotions.

There are three steps in an automatic facial expression recognition system:

- Face detection
- Facial feature extraction
- Facial expression classification

Our proposed algorithm uses four points on top, down, left and right of the face as landmarks and determines automatically the initial position for facial features based on the face height and width using a knowledge-based system.

Knowledge-based system is formed from eyes, eyebrows and mouth position on 97 subjects in Cohn-Kanade face database. Based on this system, three rectangles are located on the face as the initial position for the mouth, the left and right eyes and eyebrows as shown in Fig. 1. These rectangles are big enough to assure that they cover these features in different facial images. Additional process is used for automatically detecting of accurate feature positions and extracting of 25 FFPs. Fig. 1 shows rectangles for initial positions as well as 25 upper and lower face FFPs.

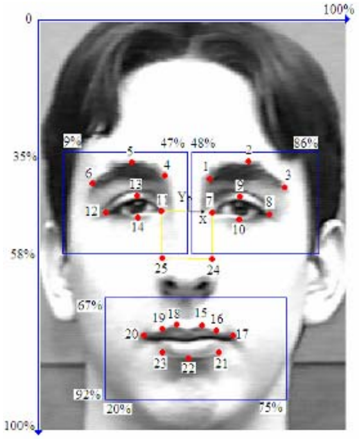


Fig. 1. Initial facial features position and 25 upper and lower face FFPs

2.1 Active Contours for Lip Localization

The active contour model algorithm, first introduced by Kass et al. [7], deforms a contour to lock onto features of interest within an image Usually the features are lines, edges, and/or object boundaries. An active contour is an ordered collection of **n** points in the image plane:

$$V = \{v_1, v_2, ..., v_n\}$$
$$v_i = (x_i, y_i), i = 1,2,...n \tag{1}$$

The points in the contour iteratively approach the boundary of an object through the solution of an energy minimization problem. For each point in the neighborhood of v_i, an energy term is computed:

$$E_i = E_{\text{int}}(v_i) + E_{ext}(v_i) \tag{2}$$

where $E_{\text{int}}(v_i)$ is an energy function dependent on the shape of the contour and $E_{ext}(v_i)$ is an energy function dependent on the image properties, such as the gradient and near point v_i.

The internal energy function used herein is defined as follows:

$$E_{\text{int}}(v_i) = cE_{con}(v_i) + bE_{bal}(v_i) \tag{3}$$

where $E_{con}(v_i)$ is the continuity energy that enforces the shape of the contour and $E_{bal}(v_i)$ is a balloon force that causes the contour to grow or shrink, **c** and **b** provide the relative weighting of the energy terms.

The external energy function attracts the deformable contour to interesting features, such as object boundaries, in an image. Image gradient and intensity are obvious characteristics to look at. The external energy function used herein is defined as follows:

$$E_{ext}(v_i) = mE_{img}(v_i) + gE_{grad}(v_i) \tag{4}$$

where $E_{img}(v_i)$ is an expression that attracts the contour to high or low intensity regions and $E_{grad}(v_i)$ is an energy term that moves the contour towards edges. Again, the constants, **m** and **g**, are provided to adjust the relative weights of the terms.

2.1.1 Two-Step Lip Active Contour

We develop a lip shape extraction and lip motion tracking system, based on a novel two-step active contours model. Four energy terms are used to control motion of control points. The points in the contour iteratively approach the outer mouth edges through the solution of a two-step energy minimization problem. One of the advantages of the proposed method is that we do not need to locate the initial snake very close to lip edges. At the first step active contour locks onto stronger upper lip edges by using both high threshold Canny edge detector and balloon energy for contour deflation. Then using lower threshold image gradient as well as balloon energy for inflation, snake inflates and locks onto weaker lower lip edges. In this stage upper control points were fixed and only lower points inflates to find lower lip edges. Fig. 2 and Fig. 3 show flowchart of proposed two- step algorithm and results.

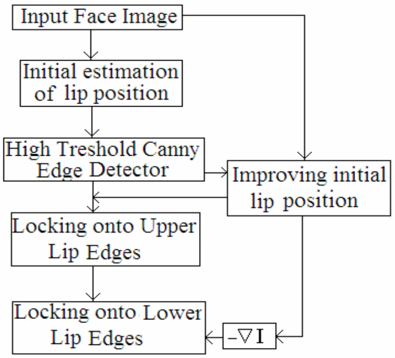

Input Face Image
Initial estimation of lip position
High Treshold Canny Edge Detector
Improving initial lip position
Locking onto Upper Lip Edges
Locking onto Lower Lip Edges
-∇I

Fig. 2. Two-step active contour algorithm

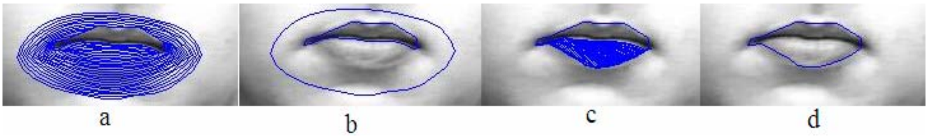


Fig. 3. Two-step lip tracking a) deflating initial snake and finding upper strong edges b) initial and final snake at the end of the first step c) fixing 14 upper control points and inflating 14 lower points to find lower weak edges d) final lip contour

In image sequences two-steps active contour is applied in the first frame (which is supposed that mouth is not open) and then the final snake is used as an initial snake in the next frame. Fig. 4 shows some results of tracking in image sequences.

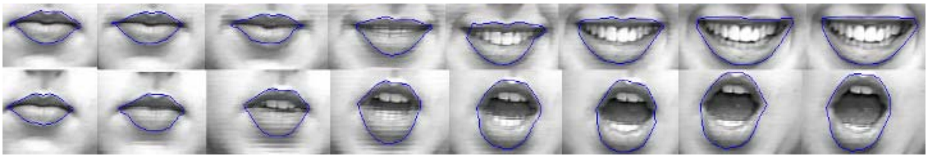


Fig. 4. Lip tracking in image sequences using proposed algorithm

2.2 Detecting FFPs in Eye and Eyebrow

We used gray-level projection method to separate eye, eyebrow and possible hair regions. Using local Min-Max methods on the gray-level, proper thresholds are determined to separate eye, eyebrow and hair regions in the initially located rectangle.

By using horizontal projection and selecting a proper threshold, eye and eyebrow regions in left and right upper faces can be separated. Also hair region is removed using vertical projection and gray level threshold for hair region. After determining eyes and eyebrows and using horizontal Sobel edge detection as well as horizontal and vertical scanning methods, 4 FFPs in eye corners and 3 FFPs in eyebrows are detected. Fig.5 shows vertical and horizontal gray-level projections, thresholds for hair region and eye-eyebrow separation and detected FFPs.

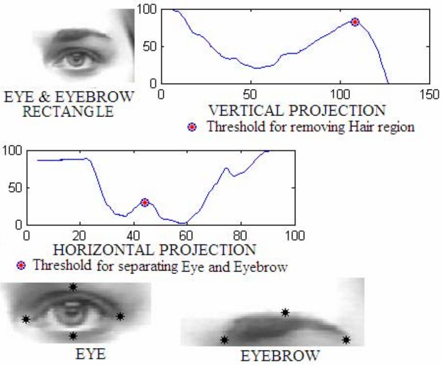


Fig. 5. Vertical and horizontal gray-level projection

3 Hybrid Tracking System

We used our enhanced two-step version of active contours to track lower face FFPs and cross correlation-based tracking system for upper face FFPs in image sequences.

Among the facial expressions, mouth has high flexibility and hard to track its shape and deformation. The use of active contours is appropriate especially when the feature shape is hard to represent with a simple template. Nevertheless, the initial active contour must be located very close to the desired feature. This problem is removed using a novel two-step active contour algorithm. In the first frame, the initial active contour is located using an estimated mouth position and lock on the outer mouth edges using two-step active contours. This contour is used as initial contour in subsequent frames (Fig. 4).

Because of openness of mouth in some static images, the proposed two-step active contour could not be applied and we used the traditional and one-step contours to detect mouth shape in static images.

Active contours method has some problems to use in upper face features. Contours are very sensitive to shadows around eyes and eyebrows.

We used a cross-correlation based tracking system for upper face features. Each upper face FFP is considered as the center of a 11×11 flow window that includes horizontal and vertical flows. Cross-correlation of 11×11 window in the first frame with a 21×21 search window at the next frame is calculated and the position by maximum cross-correlation value for two windows, were estimated as the position of the feature point for the next frame [8] [9].

Fig.6 shows detecting and tracking of upper face FFPs for surprise and disgust expressions.

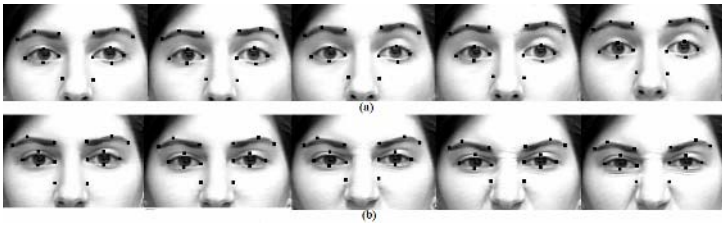


Fig. 6. Tracking of upper face FFPs a) Surprise expression b) Disgust expression

4 Feature Vector Extraction

Extracted feature points are used to extract some geometric feature points to form a feature vector for upper and lower face features.

The following features are extracted for lower face:

- Openness of mouth: average vertical distance of points 15-22 and 18-22 (Fig. 1).
- Width of mouth: horizontal distance of points 17 and 20.
- Chin rise: vertical distance of point 22 from origin.
- Lip corners distance: average vertical distance of points 17 and 20 from origin
- Normalized quadratic curvature parameters for points 15, 16 and 17.
- Normalized quadratic curvature parameters for points 17, 21 and 22.

To calculate and normalize curvature parameters, origin is transferred to point 17 that reduces curvature parameters from 3 to 2, also horizontal distance of points 17 and 22, is normalized to one.

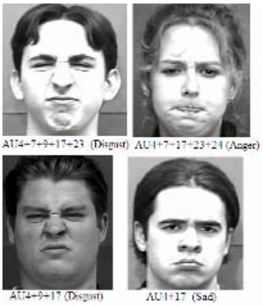
Calculated features form a 8×1 feature vector which is used for classification of lower face action units.

The following features are extracted for upper face:

- Openness of eye: vertical distance of points 9-10 and 13-14.
- Height of eyebrow 1: vertical distance of points 1 and 4 from origin.
- Height of eyebrow 2: vertical distance of points 2 and 5 from origin.
- Inner eyebrow distance: horizontal distance of points 1 and 4.
- Nose wrinkle: vertical distance of points 7-24 and 11-25.
- Normalized quadratic curvature parameters for points 1, 2 and 3.
- Normalized quadratic curvature parameters for points 4, 5 and 6.

To calculate and normalize curvature parameters, origin is transferred to point 2 and 5 in left and right eyebrow that reduces curvature parameters from 3 to 2; also horizontal distance of points 1-3 and 4-6, is normalized to one.

Points 24 and 25 are determined by a square with two vertices on points 7 and 11. These two points are used to track nose wrinkle which is a discriminant feature for disgust expression (Fig. 6). This feature can not be calculated in static images and our proposed system can not detect disgust expression in static images. As it has been shown in Fig.7, without recognizing wrinkled nose (AU9), disgust expression is very close to anger or sad expressions.



AU4+7+9+17+23 (Disgust) AU4+7+17+23+24 (Anger)

AU4+9+17 (Disgust) AU4+17 (Sad)

Fig. 7. Comparing Disgust with Sad and Anger expressions

Calculated features form a 9×1 feature vector in image sequences and a 8×1 feature vector in static images which are used for classification of upper face action units.

Mid-Point between inner eye corners is determined as origin.

In the image sequences, calculated features (except curvature parameters) in the first and the last frames are normalized using the following equation:

$$Norm\_fature = (Last\_frame - First\_frame) / First\_frame \tag{5}$$

Last frame in image sequences which represents peak of facial expressions is used to extract curvature parameters.

In the static images, distance of inner eye corners (distance of points 7 and 11 in Fig. 1) is used as a local reference to normalize extracted geometric features to remove the effect of subject-camera distance.

5 PNN Classifier

Probabilistic Neural Networks (PNN) is a variant of the Radial Basis Function Neural Networks (RBFNN) and attempts have been carried out to make the learning process in this type of classification faster than normally required for the multi-layer feed forward neural networks.

The construction of PNN involves an input layer, a hidden layer and an output layer with feed forward architecture. The input layer of this network is a set of **R** units, which accept the elements of an **R**-dimensional input feature vector. The input units are fully connected to the hidden layer with **Q** hidden units (RBF units). **Q** is the number of input/target training pairs. Each target vector has **K** elements. One of these elements is **1** and the rest are **0**. Thus, each input vector is associated with one of **K** classes.

When an input vector is presented in the input layer, the hidden layer computes distances from the input vector to the training input vectors, and produces a vector whose elements indicate how close the input is to a training input. The output layer sums these contributions for each class of inputs to produce its net output as a vector of probabilities. Finally, a compete transfer function on the output of the output layer picks up the maximum of these probabilities, and produces a **1** for that class and **0** for the other classes [9].

6 Experimental Results

The Cohn-Kanade database consists of expression sequences of subjects, starting from a neutral expression and ending with the peak of the facial expression. Subjects sat directly in front of the camera and performed a series of the facial expressions that included the six primary and also some single AUs. We used a subset of 180 image sequences containing six basic emotions for 30 subject emotions. Those AUs which are important to the communication of the emotion and were occurred at least 25 times in our database are selected. This frequency criterion ensures sufficient data for training and testing. For each person there are on average of 12 frames for each expression (after eliminating alternate frames). Image sequences for the frontal views are digitized into 640×490 pixel array with 8 bits grayscale [6].

6.1 Recognition of Upper and Lower Face AUs in Image Sequences

We used the sequence of 144 (80%) subjects as training sequences, and the sequence of the remaining 36 (20%) subjects as test sequences. This test is repeated five times, each time leaving different subjects out. The number of the input layer units in the lower face PNN classifier is equal to 8, the number of extracted features, the number of the hidden layer units equals to 144×9, the number of training pairs and that of the output layers is 9, which corresponds to selected 9 lower face AUs.

The number of the input layer units in the upper face PNN classifier is equal to 9, the number of the hidden layer units equals to144×7, and that of the output layers is 7, which corresponds to the selected 7 upper face AUs.

Table 2 shows the recognition rate of lower and upper face AUs.

Table 2. Recognition results for lower and upper face AUs in image sequences

Lower Face AUs		Upper Face AUs	
AU12	31/35 %88.57	AU1	52/65 % 80
AU15	27/29 %93.10	AU2	35/39 %89.74
AU17	73/82 %89.02	AU4	77/91 % 84.61
AU20	26/30 %86.67	AU5	30/32 %93.75
AU23	24/29 %82.76	AU6	31/38 % 81.58
AU24	23/32 %71.88	AU7	51/56 % 91.07
AU25	52/59 %88.14	AU9	30/31 % 96.77
AU26	6/10 %60.00	-	-
AU27	20/22 %90.91	-	-
Average	282/328 %85.98	Average	306/352 %86.93

Comparing to some related works [10, 11], results are encouraging.

6.2 Recognition of Six Basic Facial Emotions in Image Sequences

After classifying facial expressions into AUs, we tried to classify them to basic emotions which comprise happiness, sadness, fear, disgust, surprise, and anger.

There is no unique AUs combination for these emotions. Based on manual FACS codes for Cohn-Kanade database, a rule-base is constructed to classify facial expressions based on analyzed lower and upper face AUs. Table 3 shows this rule-bases and Table 4 shows classification results.

Table 3. Rule-bases for basic emotions classification

IF	**THEN**
(AU23=1 OR AU24 =1) AND AU9=0	Anger
AU9=1	Disgust
(AU20=1 AND AU25 =1) OR (AU20=1 AND AU26 =1)	Fear
AU12=1	Happiness
AU15=1 AND AU17 =1	Sadness
AU27=1 OR (AU5=1 AND AU26 =1)	Surprise

Table 4. Recognition rate of six basic emotions in image sequences

Anger	27/30 90%
Disgust	30/30 100%
Fear	30/30 100%
Happiness	30/30 100%
Sadness	26/30 86.67%
Surprise	30/30 100%
Average	173/180 96.11 %

Comparing to some related works [11, 12, 13], results are encouraging.

6.3 Recognition of Upper and Lower Face AUs in Static Images

Our proposed system can not detect wrinkled nose (AU9) and disgust expression in static images.

Last frame in image sequences which represents peak of facial expressions is used to train and test of static images recognition system. We left out input images for disgust expression.

We used the images of 120 (80%) subjects as training sequences, and the remaining 30 (20%) subjects as test images. This test is repeated five times, each time leaving different subjects out. The number of the input layer units in the lower face PNN classifier is equal to 8, the number of extracted features, the number of the hidden layer units equals to 120×9, the number of training pairs and that of the output layers is 9, which corresponds to selected 9 lower face AUs.

The number of the input layer units in the upper face PNN classifier is equal to 8, the number of the hidden layer units equals to120×6, and that of the output layers is 6, which corresponds to the selected 6 upper face AUs.

Table 5 shows the recognition rate of lower and upper face AUs.

Comparing to some related works [14], results are resonable.

Table 5. Recognition results for lower face AUs in static images

Lower Face AUs		Upper Face AUs	
AU12	31/35 %88.57	AU1	45/65 % 69.23
AU15	26/29 %89.66	AU2	29/39 %74.36
AU17	46/60 %76.67	AU4	43/64 % 67.19
AU20	18/30 %60.00	AU5	26/32 %81.25
AU23	18/26 %69.23	AU6	14/31 % 45.16
AU24	18/30 %60.00	AU7	24/33 % 72.73
AU25	42/55 %76.36	-	-
AU26	7/10 %70.00	-	-
AU27	22/22 %100	-	-
Average	228/297 %76.77	Average	181/264 %68.56

6.4 Recognition of Five Basic Facial Emotions in Static Images

Table 6 shows recognition results for five basic expressions (leaving out the disgust expression) using the same rule-base (Table3) and lower and upper face AUs in static images.

Table 6. Recognition rate of five basic emotions in static images

Anger	30/30 100%
Fear	23/30 76.67%
Happiness	30/30 100%
Sadness	29/30 96.67%
Surprise	29/30 96.67%
Average	141/150 94 %

7 Conclusion

In this paper we developed an automatic facial expressions analysis and classification systems with high success rate. Our image and video analysis includes automatic feature detection, tracking and the results are directly used for facial emotion classification based on AUs analysis and classification. An average recognition rate of 96.11% was achieved for six basic emotions in facial image sequences.

In the first frame, 25 Facial Feature Points (FFPs) were automatically detected, using active contours for lower face and gray level projection method for upper face. A hybrid tracking system was used to track these FFPs in subsequent frames. An enhanced version of active contour tracking system was used for lip tracking while a cross-correlation based tracking system was used to track FFPs around eyes and eyebrows.

Some geometric features were extracted, based on the position of FFPs in the first and the last frames. This features formed a feature vector which was used for classification of input image sequences into 16 AUs, using PNN. A rule-based decision making system was applied to AUs to classify input images into six basic emotions.

Proposed features and feature extraction method can also be applied to static images (except features for wrinkled nose) using a local reference to normalize these features in order to remove the effect of subject-camera distance. An average recognition rate of 94% was achieved for five basic emotions in static face images.

While most of the facial expression recognition systems use manually located FFPs in the first frame, our proposed system used automatically detection and tracking of feature points. Proposed hybrid tracking system showed robust tracking results both in upper and lower face, which only needed the rough estimated position of eye, eyebrow and mouth. Our proposed new features improved AUs recognition rate as well as six basic emotions recognition rate.

Acknowledgements. The authors appreciate the support of Iranian Ministry of Science, Research and Technology and Iranian Telecommunication Research Center (ITRC) for this research and would like to thank J.F. Cohn and T. Kanade from Pittsburgh University for their kindly providing the image database.

References

1. P. Ekman and W.V. Friesen, *FACIAL ACTION CODING SYSTEM (FACS).* Consulting Psychologists Press, Inc., (1978).
2. M. Pantic and L.J.M. Rothkrantz, “Expert system for automatic analysis of facial expressions,” Journal *of Image and Vision Computing 18(2000),* 881-905,
3. J. J. Lien, T. Kanade, J. F. Cohn and C. C. Li “ Detection, tracking and classification of action units in facial expression,“ *Journal of Robotics and Autonomous Systems 31 (2000)*, 131-146.
4. M. Valstar, I. Partas and M. Pantic,” Facial Action Unit Recognition Using Temporal Templates,” *Proc. Of the IEEE int. workshop on Robot and Human Interactive Communication*, Japan (2004), 253-258.

5. M. S. Bartlett et al.“ Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behaviour,“ *IEEE International Conference on Computer Vision and Pattern Recognition*, (2005), 568-573.
6. T. Kanade, J. Cohn, and Y. Tian. *Comprehensive database for facial expression analysis*, (2000).
7. Michael Kass, Andrew Witkin, and Demtrri Terzopoulos. “Snakes: Active contour models*,” Int. J. of Computer Vision*, (1998), 321-331.
8. H. Seyedarabi, A. Aghagolzadeh and S. Khanmohammadi, **“**Facial Expressions Animation and Lip Tracking Using Facial Characteristic Points and Deformable Model*,”. Int. Journal of Information Technology (IJIT)*, Vol.1 No. 4, (2004), 284-287.
9. Aghagolzadeh, A., Seyedarabi, H.; Khanmohammadi, S.” Single and Composite Action Units Classification in Facial Expressions by Feature-Points Tracking and RBF Neural Networks,” *Ukrainian Int. conf. signal/Image processing, UkrObraz 2004*, Kiev, Ukraine, (2004), 181-184.
10. Mohammad Yeasin, Baptiste Bullot and Rajeev Sharma, “ Recognition of Facial Expressions and Measurement of Levels of Interest From Video,” IEEE Transactions on Multimedia, vol. 8, no. 3, (June 2006), 500-508.
11. Yongmian Zhang, “ Active and Dynamic Information Fusion for Facial Expression Understanding from Image Sequences,” IEEE Transactions on pattern analysis and machine intelligence, vol. 27 no. 5, (May 2005), 699- 714 .
12. L. Ma and K. Khorasani, “ Facial Expression Recognition Using Constructive Feedforward Neural Networks,”, IEEE Transactions on Systems, Man and Cybernetics, vol.34, no.3, (June 2004), 1588-1595.
13. C. C. Chibelushi and F. Bourel, “ Hierarchical Multistream Recognition of Facial Expressions,” IEE Proceeding of Vision, Image and Signal Processing, vol. 151, no. 4, (August 2004), 303-313.
14. Maja Pantic and Leon J. M. Rothkrantz, “ Facial Action Recognition for Facial Expression Analysis From Static Face Images,” IEEE Transactions on Systems, Man and Cybernetics, vol. 34, no. 3, (June 2004) 1449-1461.

Application of 3D Co-occurrence Features to Terrain Classification

Dong-Min Woo, Dong-Chul Park, Seung-Soo Han, and Quoc-Dat Nguyen

Department of Information Engineering, Myong Ji University
Gyeonggido, Korea 449-728
dmwoo@mju.ac.kr

Abstract. Texture analysis has been efficiently utilized in the area of terrain classification. In this application, features have been obtained in the 2D image domain. This paper suggests 3D co-occurrence texture features by extending the concept of co-occurrence feature to the 3D world. The suggested 3D features are described as a 3D co-occurrence matrix by using a co-occurrence histogram of digital elevations at two contiguous positions. The practical construction of the co-occurrence matrix limits the number of levels of digital elevation. If the digital elevation is quantized into a few levels over the whole DEM (Digital Elevation Map), distinctive features cannot be obtained. To resolve this quantization problem, we employ the local quantization technique which can preserve the variation of elevations with a small number of quantization levels. Experiments are carried out using an ANN (Artificial Neural Network) classifier, and it is shown that the classification accuracy is significantly improved over the conventional classification methods with 2D features.

Keywords: texture, terrain classification, co-occurrence, 3D feature.

1 Introduction

Texture analysis has been widely used in computer vision applications, including image segmentation, image compression, and automatic inspection. Recently, it has also been employed in terrain classification using aerial and satellite imagery. This particular application is significantly important from the viewpoint of resource management, environment preservation, and national defense.

Texture is a kind of spatial distribution of gray-level variations or regular structural patterns in an image. [1,2] Major properties of texture include coarseness, contrast, directionality, line-likeness, regularity and roughness. [1] Texture features reflecting these properties have been suggested by using co-occurrence [3], MRF(Markov Random Field) [4], Garbor filter [5], Fractal [6], etc. Among these texture features, the co-occurrence feature was reported to be the most effective for terrain classification. [7]

3D texture introduced by Dana [8] and Wang [9] considers the physical characteristics of an object surface in the real world. The addition of 3D texture features

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 305–313, 2006.

can thus improve the accuracy of terrain classification. However, these early 3D features do not directly reflect 3D texture from the physical appearance of the surface. In this paper, we propose a new 3D co-occurrence feature, which directly and systematically defines 3D texture from a DEM (Digital Elevation Map). In computing the co-occurrence feature, implementation of a co-occurrence matrix requires quantization of elevation with several levels. A quantization scheme such as histogram equalization with several levels can preserve texture information in 2D image. In a DEM, however, the dynamic range of elevation change is so wide that it is not possible to obtain texture information from the elevation quantized in a general way. In the present paper, in order to preserve the texture information of quantized elevation, we employ the local quantization scheme. Since quantization is carried out locally, we can obtain the texture information with only a few quantization levels.

2 3D Co-occurrence Feature

2.1 Generation of DEM

To generate the DEM, we perform area-based stereo matching with the Terrest system [10], developed at the University of Massachusetts and Myongji University. The goal of stereo image matching in the Terrest system is to find a disparity map $D_i(i,j)$ that maps the pixels in an epipolar resampled reference image $I_R(i,j)$ into an epipolar resampled warped image $I_W(i,j)$ such that each pixel pair sees the same spot on the object, i.e., $I_R(i,j)$ and $I_W(i+D_i(i,j),j)$ view the same spot on the surface. To find the accurate disparity map, we employed NCC (Normalized Cross-Correlation) [11] and a multi-resolution scheme [12], referred to as hierarchical, or pyramid processing, in the Terrest system.

From the disparity map obtained via stereo matching, 3D coordinates are calculated by 3D triangulation of corresponding points. To generate the DEM as a 3D terrain model, we obtain the elevation for each ortho-rectified grid. The elevation can be calculated by interpolating the neighboring 3D coordinates.

2.2 Computation of 3D Co-occurrence Feature

To extract 3D co-occurrence feature from DEM, we employ a similar procedure to 2D co-occurrence feature extraction suggested by Haralick [3]. The co-occurrence feature is used to evaluate the spatial dependency in terms of a co-occurrence matrix. The co-occurrence matrix is defined as a matrix, the elements of which represent the number of occurrences that elevation level i deviates from elevation level j by a prescribed distance and angle. In this paper, we use a unit distance and four angles of $\theta = 0^o, 45^o, 90^o, 135^o$ and the co-occurrence matrix can be specified as $P_{ij\theta}$.

Because the dimension of the co-occurrence matrix is the number of quantization levels squared GxG, this calculation tends to be computationally expensive if the quantization level G is high. Thus, realistic implementation of the co-occurrence matrix requires a few levels of quantization, such as 8 levels in the common calculation of a 2D co-occurrence matrix. Unfortunately, a small number of quantization levels obviously removes most of texture information from the quantized elevation data.

In this context, we employ a local quantization scheme which can preserve texture information with a small number of quantization levels. This scheme begins with the estimation of the plane from the elevation data in a local area. If we quantize the deviation of each elevation from the fitted plane, we can minimize the loss of texture information, as shown in Fig. 1.

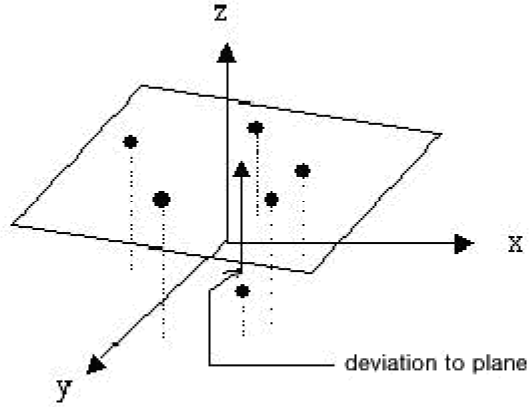


Fig. 1. Local quantization of elevation data

The locally fitted plane equation is specified as z = ax + by + c. To estimate the coefficients, a, b, and c, we employ a matrix equation (1), which is obtained by substitution of n local elevation data, (x_1, y_1, z_1), (x_2, y_2, z_2), ... , (x_n, y_n, z_n).

$$Ax = b \tag{1}$$

where rows of matrix A are $(x_i,\ y_i,\ 1)$ for i=1, ..., n, and $x = (a, b, c)^t$, $b = (z_1, z_2, ..., z_n)^t$.

The least squared error estimation of the coefficient vector x can be evaluated by

$$x = (A^t A)^{-1} A^t b. \tag{2}$$

The quantized value for any elevation, z_i, is the deviation of the elevation from the fitted plane, given by

$$d_i = z_i - (x_i, y_i, 1)x. \tag{3}$$

The local elevation data used to find the fitted plane can be within a small window in the DEM. Since this window is centered on the position where the deviation is calculated, its window size does not significantly affect the resultant 3D co-occurrence features. In this paper, we use a 5x5 window in consideration of the computational burden.

The employed 3D co-occurrence features are ASM (Angular Second Moment), CON (Contrast) and ENT (Entropy). ASM measures the homogeneity of the elevation data, given by

$$f_{i\theta} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P_{ij\theta}^2 \tag{4}$$

Since the homogeneity of the elevation indicates the flatness, a high ASM, as in the road surface, is obtained. CON represents contrast or partial variation of the elevation data, given by

$$f_{2\theta} = \sum_{n=0}^{G-1} n^2 \sum_{|i-j|=n} P_{ij\theta} \tag{5}$$

Since CON is high in areas with high variation, a significantly high CON is obtained in a foliage or bush area with many trees. ENT provides the measure of the complexity and is computed by an entropy equation, given by

$$f_{3\theta} = -\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P_{ij\theta} \ln(P_{ij\theta}) \tag{6}$$

ENT is high in areas with complex and random elevations.

3 ANN Classifier

In this paper, an ANN (Artificial Neural Network) classifier has been used for the terrain classification. The employed neural network algorithm is based on the MLPNN (Multi-Layered Perceptron Neural Network) type [13].

MLPNN generally consists of one input layer, one or more hidden layers, and one output layer. Each layer is constructed with a number of neurons, as shown in Fig. 2. Each neuron is connected with neurons from the previous layer with appropriate weights. The weights are updated in weight space using the method of gradient descent, so that the error between the output and the target can be decreased. A weight update is also carried out by

$$W_{ji}(new) = W_{ji}(old) + \varDelta W_{ji}, \tag{7}$$

where W_{ji} is a weight connecting the ith neuron to the jth neuron. The increment of the weight is expressed as

$$\varDelta W_{ji} = \alpha \delta_i O_j, \tag{8}$$

$$\delta_i = (t_i - O_i) O_i (1 - O_i), \tag{9}$$

$$\delta_i = O_i (1 - O_i) \sum_k \delta_i W_{ki}. \tag{10}$$

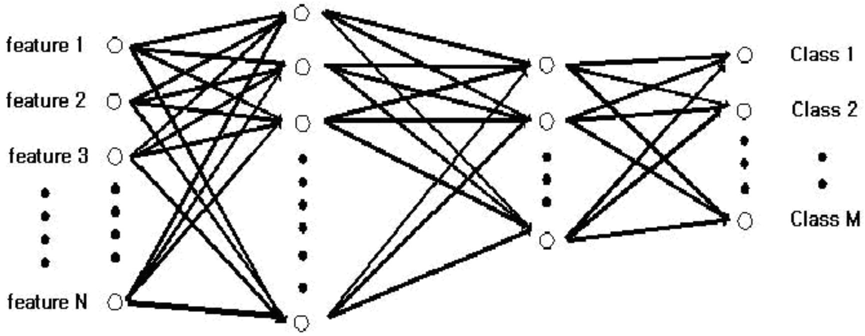


Fig. 2. Structure of ANN classifier

O_i is the neuron output value at each layer, t_i is the target value, and α is learning gain. For the evaluation of δ_i, equation (9) is used at the output layer, while equation (10) is used for all other layers.

The ANN-based classifier adopted in this paper has a Nx15x5xM structure, as shown in Fig. 2. Note that N and M represent the number of texture features and the number of classes, respectively.

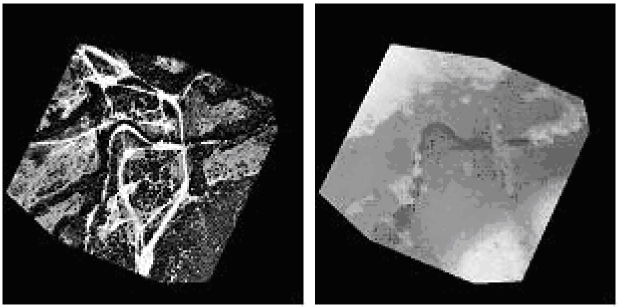
4 Experimental Results

The experimental environment was set up so as to classify aerial image data into 4 classes: foliage, grass-covered ground, bare ground, and shadow. Four feature sets are used for the experiments. Feature set A includes only 2D co-occurrence features, and feature set B includes image intensity and 2D co-occurrence features. Feature sets C and D are produced by the addition of 3D co-occurrence features to feature sets A and B, respectively.

The ortho-image and DEM generated by aerial images are shown in Fig. 31 and Fig. 32, respectively. Fig. 41 shows the ground truth, where the white area represents bare ground such as roads, the light gray area is for foliage, the dark gray area is for grass-covered ground, and the black area represents shadows. The first step of this experiment is to extract the training data for the classifier. We randomly selected the training area, which is 1% of each class, shown in small windows as in Fig. 42.

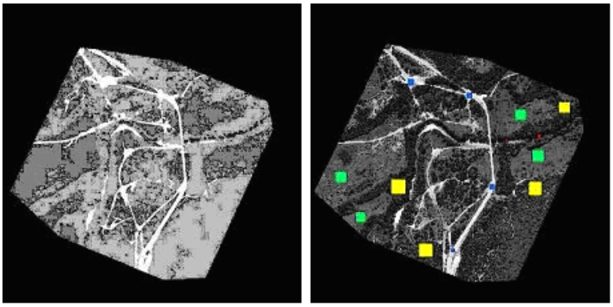
To calculate 3D co-occurrence features, first the 3D co-occurrence matrix should be established. In constructing the 3D co-occurrence matrix, we carried out local quantization with 8 quantization levels, which yields an 8x8 3D co-occurrence matrix. Since we use three types of co-occurrence features - ASM, CON, ENT - for 4 angular directions, 12 3D co-occurrence features are calculated.

For the experiments, 2D co-occurrence features must to be calculated, similarly. To construct an 8x8 2D co-occurrence matrix, an ortho-image with 8 gray levels is needed. Histogram equalization was carried out to obtain this image. Three types of 2D co-occurrence features are calculated using the same



1 Ortho-Image 2 DEM

Fig. 3. Ortho-image and DEM used in the terrain classification



1 ground truth 2 training data

Fig. 4. Ground truth of experimental terrain and its training data

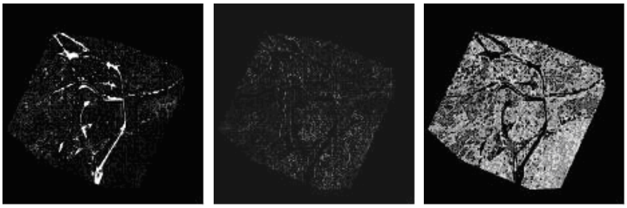
procedure as delineated by equations (4), (5) and (6). Fig. 5 and Fig. 6 show ASM, CON and ENT features with $\theta = 0^o$ in gray scale for 3D co-occurrence and 2D co-occurrence, respectively. Since 3D co-occurrence features are evaluated in terms of physical appearance, not just brightness of pixel, they are significantly different from 2D co-occurrence features.

To carry out the classification experiments based on feature sets A, B, C and D, we implemented four ANN-based classifiers. Since there are four terrain classes, the classifier has an Mx15x5x4 structure. M represents the number of texture features, and depends on the selection of the feature set. Table 1 presents the classification results using 2D co-occurrence features with or without image intensity, where the bold number indicates the number of correctly classified pixels for each class. In this case, the addition of image pixel intensity significantly affects the classification accuracy. With feature set A, which uses only 12 co-occurrence features, the error rates are very high except for foliage classification.

Table 2 shows classification results using both 2D and 3D co-occurrence features. In comparison with the results given in Table 1, the addition of 3D co-occurrence features improves the classification accuracy. In particular, the classification of road and foliage is significantly improved. This is due to the use

1 3D ASM 2 3D CON 3 3D ENT

Fig. 5. 3D co-occurrence features



1 2D ASM 2 2D CON 3 2D ENT

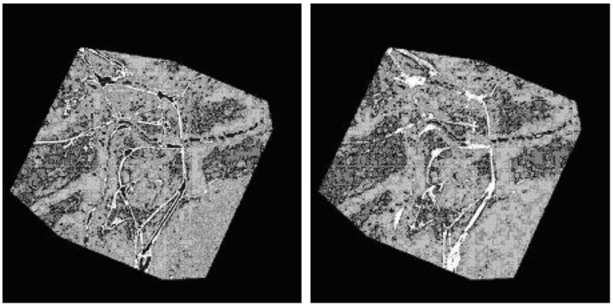
Fig. 6. 2D co-occurrence features

Table 1. Classification result using 2D co-occurrence features (unit:1,000 pixels)

ground truth	Feature Set A				Feature Set B			
	Shadow	grass	foliage	road	shadow	Grass	foliage	road
shadow(41.8)	**0.258**	0.778	2.768	3.799	**26.193**	0.707	9.187	6.602
grass(683.0)	0.319	**1.569**	0.787	1.284	0.149	**567.594**	219.001	34.121
foliage(1018.6)	27.687	661.416	**1007.673**	82.590	14.656	94.550	**784.581**	1.238
road (193.4)	13.553	19.243	7.343	**105.730**	0.805	20.155	5.735	**151.422**
total(1936.8)	41.817	683.006	1018.571	193.403	41.803	683.006	1018.504	193.383
	correctly classified: 1115.23 (57.58%)				correctly classified: 1529.79 (78.99%)			

Table 2. Classification result with the addition of 3D co-occurrence features(unit:1,000 pixels)

ground truth	Feature Set A				Feature Set B			
	shadow	grass	foliage	road	shadow	Grass	foliage	road
shadow(41.8)	**9.826**	1.905	20.204	1.528	**20.936**	0.522	3.968	0.156
grass(683.0)	0.109	**492.786**	100.706	13.638	0.009	**513.964**	69.458	26.371
foliage(1018.6)	31.881	147.978	**884.771**	3.791	20.872	139.353	**939.035**	3.710
road (193.4)	0.001	26.008	13.961	**172.475**	0.000	29.167	6.110	**163.166**
total(1936.8)	41.817	668.677	1019.642	191.432	41.817	683.006	1018.571	193.403
	correctly classified: 1559.858 (80.54%)				correctly classified: 1637.101 (84.52%)			



1 feature set B 2 feature set D

Fig. 7. Classification results

of the physical surface characteristics of the real world, thus indicating that the suggested 3D co-occurrence features can be utilized very efficiently in terrain classification applications.

Fig. 71 shows the classification result using pixel intensity and 2D co-occurrence features (feature set B), and Fig. 72 shows the classification result using pixel intensity, 2D co-occurrence features and 3D co-occurrence features (feature set D). In comparison with the ground truth as in Fig. 4, we find that the addition of 3D co-occurrence features improves overall classification accuracy. In particular, the classification between road and shadow is distinctively improved, due to the addition of 3D co-occurrence features.

5 Conclusions

In this paper we have proposed the use of 3D co-occurrence features, which can effectively reflect physical surface characteristics in the real world in a direct fashion, for the purpose of terrain classification. Experimental results show that the addition of 3D co-occurrence features significantly improves classification accuracy. However, since classified ground truth is relatively scarce, experiments were carried on a single aerial image set. In this context, extensive experiments involving various sites with classified ground truths, in conjunction with intensive analyses of the effects of 3D co-occurrence features should be carried out in future work.

References

1. Tamura, H., Mori, S., Yamawaki, T.: Textural Features Corresponding to Visual Perception. IEEE Trans. Systems, Man and Cybernetics, Vol. 8 (1978) 460-473
2. Haralick, R.: Statistical and Structural Approaches to Texture. Proceeding IEEE, Vol. 67 (1979), 786-804
3. Haralick, R., Shanmugam, K., Dinstein, I.: Texture Features for Image Classification. IEEE Trans. Systems, Man and Cybernetics, Vol. 3 (1973) 610-621
4. Dubes, R., Jain, A.: Random Field Models in Image Analysis. Journal of Applied Statistics, Vol. 16 (1989) 131-164

5. Campbell, F., Robson, J.: Application to Fourier Analysis to the Visibility of Gratings. Journal of Physiology, Vol. 197 (1968) 551-566
6. Kube., P., Pentland, A.: On the Imaging of Fractal Surfaces. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 10 (1988) 704-707
7. Ohanian, P., Dubes, R.: Performance Evaluation for Four Classes of Textural Features. Pattern Recognition, Vol. 25 (1992) 819-833
8. Dana, K., Nayar, S., Van Ginneken, B., Koenderink, J.: Reflectance and Texture of Real-World Surfaces. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (1997) 151-157
9. Wang, X., Stolle, F., Schultz, H., Riseman, E., Hanson, A.: Using Three-Dimensional Features to Improve Terrain Classification. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (1997) 915-920
10. Schultz, H.: Terrain Reconstruction from Widely Separated Images. Proceeding of SPIE, Vol. 2486 (1995) 113-122
11. Panton, D. J.: A Flexible Approach to Digital Stereo Mapping. Photogrammetric Engineering and Remote Sensing, Vol. 44 (1978) 1499-1512
12. Hannah, M.: A System for Digital Stereo Image Matching. Photogrammetric Engineering and Remote Sensing, Vol. 55 (1989) 1765-1770
13. Fausett, L.: Back-Propagation Neural Net. Fundamentals of Neural Network, Prentice Hall (1994) 289-333

$(2D)^2$ DLDA for Efficient Face Recognition

Dong-uk Cho[1], Un-dong Chang[2], Kwan-dong Kim[2],
Bong-hyun Kim[3], and Se-hwan Lee[3]

[1] Department of Information & Communication Engineering
Chungbuk Provincial University of Science & Technology, Chungbuk, Korea
ducho@ctech.ac.kr
[2] Department of Computer Engineering
Chungbuk National University, Chungbuk, Korea
udchang@naver.com
[3] Department of Computer Engineering
Hanbat National University, Daejeon, Korea
{bhkim, sian}@hanbat.ac.kr

Abstract. In this paper, a new feature representation technique called 2-directional 2-dimensional direct linear discriminant analysis ($(2D)^2$ DLDA) is proposed. In the case of face recognition, the small sample size problem and need for many coeffficients are often encountered. In order to solve these problems, the proposed method uses the direct LDA and two directional image scatter matrix. The ORL face database is used to evaluate the performance of the proposed method. The experimental results show that the proposed method obtains better recognition rate and requires lesser memory than the direct LDA.

Keywords: Linear Discriminant Analysis, Direct LDA, Face Recognition.

1 Introduction

Nowadays, Face recognition has been an active research. Various methods have been proposed for Face recognition. Especially, the appearance-based methods have been successfully employed. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are well known methods among them. The PCA seeks directions that have the largest variance associated with it. On the other hand, the LDA seeks directions that are efficient for discrimination between classes.

In general, it is believed that LDA-based pattern classification methods outperform PCA-based ones. However, The traditional LDA has small sample size (SSS) problem. The SSS problem arises when the number of training samples is smaller than the dimensionality of the samples[1]. Also, it is difficult to directly apply the LDA to high dimensional matrix because of computational complexity. To solve the problem, Belhumeur et al. [2] proposed Fisherfaces method based on LDA. They proposed dimensionality reduction using PCA before LDA. But it has a potential problem. It is that PCA step may discard dimensions that contain important discriminative information. Chen et al. [3] proved that the null space of within-class scatter matrix contains the most discriminative information. In reality, PCA discards

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 314 – 321, 2006.

the null space of the within-class scatter matrix. Therefore, in order to prevent the null space from discarding, Yu et al. [4] proposed direct LDA (DLDA) method. the DLDA directly processes data in the original high dimensional vectors. By simultaneous diagonaliza-tion, the DLDA is able to discard the null space of between-class scatter matrix and to keep the null space of within-class scatter matrix, which contains very important discriminative information. J. Lu et al. [5] proposed kernel direct discriminant analy-sis (KDDA). While the KDDA provides better performance, it is computationally more than the DLDA

The PCA based methods have been developed since the eigenfaces methods[6,7] was presented for face recognition. Recently, Yang et al. [8] proposed two dimensional PCA (2DPCA). While previous methods use 1D image vector, the 2DPCA makes directly the scatter matrix from 2D image matrices. The 2DPCA deals with the small size scatter matrix than the traditional PCA-based methods and evaluates the scatter matrix accurately. For example, an image vector of 112×92 forms 10304 dimensional vector and the size of the scatter matrix is 10304×10304. On the other hand, the covariance of the 2DPCA forms only 92×92 matrix. Also, the 2DPCA is more suitable for small sample size problems because its scatter matrix is small. But it requires more coefficients for image representation than PCA. Therefore it needs more storage and more time for recognition. L. Wang et al. [9] showed that the 2DPCA is equivalent to a special case of the block-based PCA. Specially, the blocks are the row directional lines of the images.

In this paper, we introduce a new low-dimensional feature representation method, $(2D)^2$ DLDA. The proposed method makes the row directional and the column directional image scatter matrix by considering the row and column directional lines of the image respectively. The image scatter matrix reduces the chance of singularity caused by SSS problem. And then the DLDA algorithm is used for obtaining the feature matrix. It maximizes Fisher's criterion.

The remainder of this paper is organized as follows. In Section 2, the proposed $(2D)^2$ DLDA algorithm is described. Experimental results and comparisons with the DLDA is presented in Section 3. Finally, conclusions are offered in Section 4.

2 $(2D)^2$ DLDA

2.1 Row Directional 2D DLDA

Let X denotes a $m \times n$ image, and W is an n-dimensional column vector. X is projected onto W by the following linear transformation

$$Y = XW \tag{1}$$

Thus, we get an m-dimensional projected vector Y, called the feature vector of the image X. Suppose there are C known pattern classes in the training set, and M denotes the size of the training set. The jth training image is denoted by a $m \times n$ matrix X_j ($j = 1, 2, \cdots, M$) and the mean image of all training sample is denoted by

$\overline{X}$ and $\overline{X}_i (i = 1, 2, \cdots, c)$ denoted the mean image of class T_i and N_i is the number of samples in class T_i, the projected class is P_i. After the projection of training image onto W, we get the projected feature vector

$$Y_j = X_j W, \quad j = 1, 2, \cdots, M \tag{2}$$

LDA attempts to seek a set of optimal discriminating vectors to form a transform W by maximizing the Fisher criterion denoted as

$$J(W) = \frac{tr(\tilde{S}_b)}{tr(\tilde{S}_w)} \tag{3}$$

Where tr(·) denotes the trace of matrix, $\tilde{S}_b$ denotes the between-class scatter matrix of projected feature vectors of training images, and $\tilde{S}_w$ denotes the within-class scatter matrix of projected feature vectors of training images. So,

$$\begin{aligned} \tilde{S}_b &= \sum_{i=1}^{C} N_i (\overline{Y}_i - \overline{Y})(\overline{Y}_i - \overline{Y})^T = \sum_{i=1}^{C} N_i [(\overline{X}_i - \overline{X})W][(\overline{X}_i - \overline{X})W]^T, \\ \tilde{S}_w &= \sum_{i=1}^{C} \sum_{Y_k \in P_i} (Y_k - \overline{Y}_i)(Y_k - \overline{Y}_i)^T = \sum_{i=1}^{C} \sum_{X_k \in T_i} [(X_k - \overline{X}_i)W][(X_k - \overline{X}_i)W]^T \end{aligned} \tag{4}$$

So,

$$\begin{aligned} tr(\tilde{S}_b) &= W^T \left(\sum_{i=1}^{C} N_i (\overline{X}_i - \overline{X})^T (\overline{X}_i - \overline{X}) \right) W, \\ tr(\tilde{S}_w) &= W^T \left(\sum_{i=1}^{C} \sum_{X_k \in T_i} (X_k - \overline{X}_i)^T (X_k - \overline{X}_i) \right) W \end{aligned} \tag{5}$$

Let us define the following matrix

$$R_b = \sum_{i=1}^{C} N_i (\overline{X}_i - \overline{X})^T (\overline{X}_i - \overline{X}), \; R_w = \sum_{i=1}^{C} \sum_{X_k \in T_i} (X_k - \overline{X}_i)^T (X_k - \overline{X}_i) \tag{6}$$

The matrix R_b is called the row directional image between-class scatter matrix and R_w is called the row directional image within-class scatter matrix.

Alternatively, the criterion can be expressed by

$$J(W_r) = \frac{W_r^T R_b W_r}{W_r^T R_w W_r} \tag{7}$$

Now, we try to find a matrix that simultaneously diagonalizes both R_b and R_w.

$$A R_w A^T = I, \; A R_b A^T = \Lambda_r \tag{8}$$

Where Λ_r is a diagonal matrix with diagonal elements sorted in decreasing order.

First, we find eigenvectors V_r that diagonalizes R_b

$$V_r^T R_b V_r = \Lambda_r \tag{9}$$

Where $V_r^T V_r = I$. Λ_r is a diagonal matrix sorted in decreasing order, i.e. each column of V_r is an eigenvector of R_b, and Λ_r contains all the eigenvalues.

Let Y_r be the first l columns ($l \le n$) of V_r (a $n \times n$ matrix, n being the column numbers of image).

$$V_r^T R_b Y_r = D_b \tag{10}$$

Where D_b is the $l \times l$ principal sub-matrix of Λ_r.

Further let $Z_r = Y_r D_b^{-1/2}$ to unitize R_b,

$$(Y_r D_b^{-1/2})^T R_b (Y_r D_b^{-1/2}) = I \Rightarrow Z_r^T R_b Z_r = I \tag{11}$$

Next, we find eigenvectors U_r to diagonalize $Z_r^T R_w Z_r$.

$$U_r^T Z_r^T R_w Z_r U_r = D_w \tag{12}$$

Where $U_r^T U_r = I$. D_w may contain zeros in its diagonal.

To maximize $J(W_r)$, we can sort the diagonal elements of D_w and discard some high eigenvalues with the corresponding eigenvectors.

Let the optimal projection matrix, W_r

$$W_r = (D_w^{-1/2} U_r^T Z_r^T)^T \tag{13}$$

Also, W_r unitizes R_w [6,8].

2.2 Column Directional 2D DLDA

Let us define the following matrix

$$C_b = \sum_{i=1}^{C} N_i (\overline{X}_i - \overline{X})(\overline{X}_i - \overline{X})^T,\ C_w = \sum_{i=1}^{C} \sum_{X_k \in T_i} (X_k - \overline{X}_i)(X_k - \overline{X}_i)^T \tag{14}$$

The matrix C_b is called the column directional image between-class scatter matrix and C_w is called the column directional image within-class scatter matrix.

Alternatively, the criterion can be expressed by

$$J(W_c) = \frac{W_c^T C_b W_c}{W_c^T C_w W_c} \tag{15}$$

Now, we try to find a matrix that simultaneously diagonalizes both C_b and C_w.

$$BC_wB^T = I,\ BC_bB^T = \Lambda_c \tag{16}$$

Where Λ_c is a diagonal matrix with diagonal elements sorted in decreasing order.

First, we find eigenvectors V_c that diagonalizes C_b

$$V_c^T C_b V_c = \Lambda_c \tag{17}$$

Where $V_c^T V_c = I$. Λ_c is a diagonal matrix sorted in decreasing order, i.e. each column of V_c is an eigenvector of C_b, and Λ_c contains all the eigenvalues.

Let Y_c be the first k columns ($k \le m$) of V_c (a $m \times m$ matrix, m being the row numbers of image).

$$V_c^T C_b Y_c = \tilde{D}_b \tag{18}$$

Where $\tilde{D}_b$ is the $k \times k$ principal sub-matrix of Λ_c.

Further let $Z_c = Y_c \tilde{D}_b^{-1/2}$ to unitize C_b,

$$(Y_c \tilde{D}_b^{-1/2})^T C_b (Y_c \tilde{D}_b^{-1/2}) = I \Rightarrow Z_c^T C_b Z_c = I \tag{19}$$

Next, we find eigenvectors U_c to diagonalize $Z_c^T C_w Z_c$.

$$U_c^T Z_c^T C_w Z_c U_c = \tilde{D}_w \tag{20}$$

Where $U_c^T U_c = I$. $\tilde{D}_w$ may contain zeros in its diagonal.

To maximize $J(W_c)$, we can sort the diagonal elements of $\tilde{D}_w$ and discard some high eigenvalues with the corresponding eigenvectors.

Let the optimal projection matrix, W_c

$$W_c = (\tilde{D}_w^{-1/2} U_c^T Z_c^T)^T \tag{21}$$

Also, W_c unitizes C_w.

2.3 (2D)^2 DLDA

As we discussed in Section 2.1 and 2.2, row directional 2D DLDA and column direction 2D DLDA produce optimal projection matrix W_r and W_c, respectively. To project an $m \times n$ image X onto W_r yields m by l matrix $Y_{m \times l} = X_{m \times n} \cdot W_{n \times l}$. Similarly, to project an $m \times n$ image X onto W_c yields a k by n matrix $Y_{k \times n} = W_{m \times k}^T \cdot X_{m \times n}$.

Suppose that we project the $m \times n$ image X onto W_r and W_c simultaneously, we obtain a k by l matrix X^*,

$$X^* = W_c^T (X - \overline{X}) W_r . \tag{22}$$

The distance measure to classify two matrices is the nearest neighbor, The distance between X_1^* and X_2^* is adopted by Frobenius norm. The Frobenius norm as follows

$$D_F(X_1^*, X_2^*) = \left\| X_1^* - X_2^* \right\|_F \tag{23}$$

3 Experimental Results

The proposed method is tested on the ORL face image database (http://www.cam-orl.co.uk/facedatabase). The ORL database consists of 40 distinct persons. There are 10 images per person. The images are taken at different times and contain various facial expressions (open/closed eyes, smiling/ not smiling) and facial details (glasses or no glasses). The size of image is 92×112 pixels with 256 gray levels. Fig. 1 depicts some sample images in the ORL database. Five sets of experiments are conducted. In all cases the five images per class are randomly chosen for training from each person and the other five images are used for testing. Thus the total number of training images and testing images are both 200. All of our tests are carried out on a PC with P4 1.5 GHz CPU and 512MB RAM memory. To simulate algorithm, matalb 6 platform is used.

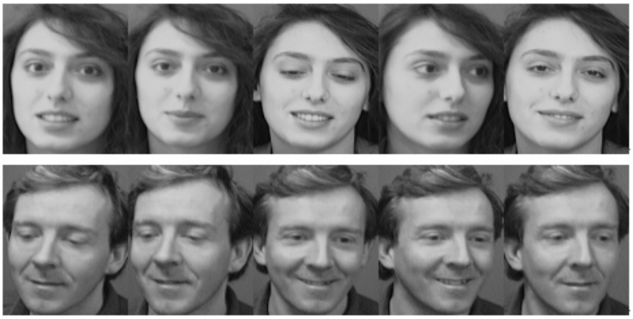


Fig. 1. Some face samples of ORL face database

Table 1 compares the average recognition rates and the dimension size obtained using the (2D)^2 DLDA, the row directional 2D DLDA, the column directional 2D DLDA and the DLDA. 2D based methods have a merit that the recognition rate is high. Whereas they also have a weak point that the dimension size of feature matrix is larger than 1D based methods. However the proposed method has not only high recognition rate but also the small dimension size. In the 112×92 image matrix, the size of the row directional image scatter matrix and the size of the column directional image scatter matrix is 92×92 and 112×112, respectively.

Table 1. Comparison of average recognition rates of different methods

Methods	Average Recognition rate(%)	Dimension
DLDA	90.6	40
Row directional 2D DLDA	94.1	112×7
Column directional 2D DLDA	94.1	8×92
$(2D)^2$ DLDA	94.5	8×7

In the $(2D)^2$ DLDA, the row directional image scatter matrix and the column directional image scatter matrix is used, simultaneously. As a result, the feature matrix size is much smaller and the recognition rate is higher than the row directional 2D DLDA or the column directional 2D DLDA. Table 1 shows that the average recognition rate of the $(2D)^2$ DLDA is higher than other methods and the dimension size is small like DLDA.

4 Conclusion

In this paper, the $(2D)^2$ DLDA algorithm is proposed. The method combines the merits of the image scatter matrix and the DLDA approaches. Since the size of the image scatter matrix is smaller than the conventional method, SSS problem can be avoided and eigenvectors can be efficiently computed. Furthermore it achieves the better performance by using the DLDA since the DLDA preserves the null space of within-class scatter matrix, which contains very important discriminative information. Also, to obtain the low dimensional feature matrix, we project image matrix onto the row directional and the column directional projection matrix, simultaneously. The experimental results show that the average recognition rate of the $(2D)^2$ DLDA is higher than other methods and the dimension size is less than other 2D based methods.

References

1. Fukunaga, K.: Introduction to Statistical Pattern Recognition. 2nd edn. Academic Press, New York (1990)
2. Belhumeour, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Analysis and Machine Intelligence. Vol. 19. No. 7. (1997) 711-720
3. Chen, L., Liao, H., Ko, M., Lin, J., Yu, G.: A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition. Vol. 33. No. 10. (2000) 1713-1726

4. Yu, H., Yang, J.: A direct LDA algorithm for high-dimensional data with application to face recognition. Pattern Recognition. Vol. 34. No. 10. (2001) 2067-2070
5. Lu, J., Plataniotis, K. N., Venetsanopoulos, A. N.: Face Recognition Using Kernel Direct Discriminant Analysis Algorithms. IEEE Trans. Neural Networks. Vol. 14. No. 1. (2003) 117-126
6. Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (1994) 84-91
7. Turk, M., Pentalnd, A.: Eigenfaces for recognition. J. Cognitive Neurosci. Vol. 3. No. 1. (1991) 71-86
8. Yang, J., Zhang, D., Frangi, A.F., Yang, J.: Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence. Vol. 26. No. 1. (2004) 131-137
9. Wang, L., Wang, X., Zhang, X., Feng, J.: The equivalence of two-dimensional PCA to line-based PCA. Pattern Recognition Letters. Vol. 26. (2005) 57-60

Automatic Skull Segmentation and Registration for Tissue Change Measurement After Mandibular Setback Surgery

Jeongjin Lee[1], Namkug Kim[2,*], Ho Lee[1], Suk-Ho Kang[2],
Jae-Woo Park[3], and Young-Il Chang[3]

[1] School of Electrical Engineering and Computer Science, Seoul National University,
San 56-1 Shinlim 9-dong Kwanak-gu, Seoul 151-742, Korea
leejeongjin@gmail.com, holee@cglab.snu.ac.kr
[2] School of Industrial Engineering, Seoul National University
namkugkim@gmail.com, shkang@cybernet.snu.ac.kr
[3] Department of Orthodontics, School of Dentistry and Dental Research Institute, Seoul National University
jeuspark@empal.com, drchang@plaza.snu.ac.kr

Abstract. In this paper, we propose an automated system that registers dental CT scans at pre- and post-operative states for a three-dimensional analysis on soft and hard tissue changes after mandibular setback surgery. Our registration method matches automatically extracted skulls to obtain optimal registration parameters based on the rigid transformation. Chamfer distance map algorithm is employed to accelerate a registration speed by referring to pre-calculated distance value and eliminating burdens of point-to-point correspondence identification. Skull surface registration corrects the translational and rotational mismatch. During an adaptive optimization, search range and step are dynamically changed to achieve finer alignments fast and robustly. Our method has been successfully applied to eight pairs of pre- and post-operative CT scans. Experimental results show that our algorithm is more accurate, and converges faster than conventional ones. Using a grid measurement, the changes of bone, and soft tissue were measured in skeletal Class III mandibular prognathism patients. Our method could be applicable to the other oral and maxillofacial surgeries as well as plastic surgeries.

1 Introduction

Recently, with advancements in orthognathic surgical techniques, surgery cases have increased, including those performed to correct the underlying skeleton in Class III patients. Class III mandibular prognathism is one of dentofacial deformity, which the mandible is in the anterior position to maxilla compared to the normal relationship. The consequent facial appearance is of great importance, even when the patient's chief purpose in treatment is not concerned with cosmetics. A more accurate prediction of

[*] Corresponding author.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 322 – 331, 2006.

the surgical result comprises an essential part of the diagnosis and treatment planning of orthognathic surgery.

Hershey and Smith [1] have shown that soft tissue changes could be predicted from the skeletal changes, according to the interplay of the cephalometric landmarks of the hard and soft tissue profiles. The evaluation and prediction of surgical treatment was usually performed by superimposition of cephalometric tracings. One of the most popular superimposition methods is the best fit of the cranial base anatomy. The cranial base is considered a satisfactory reference for cephalometric superimposition, since it grows rapidly in early postnatal life [2-3]. The use of reference line has been reported to be relatively stable. SN line, which is drawn by the intersection of S(sella) and Na(nasion) points, is frequently used as a reference line [4-5]. Steiner [5] used SN with registration point at S to evaluate sagittal changes in mandibular positions and at Na to evaluate the position of the maxilla. However, all of these methods were limited to two-dimensional assessments.

After surgery, the facial soft tissue was actually altered on all three dimensions, which caused a significant difference between the prediction and the surgical result [6]. McCance et al. [7] tried to analyze the soft tissue changes after surgery in three dimensions using a laser scan. Moss et al. [8] suggested that the laser scan could be an effective tool to evaluate the three dimensional changes after orthodontic treatment. However, the laser scan could not reveal the relations between the soft tissue and the underlying hard tissue. McCance et al. [9] investigated the soft tissue changes after orthognathic surgery using a CT scan. He concluded that the radial measurement from the center of rotation of the head could not be directly comparable to the linear measurements on a 2D lateral cephalometric radiograph.

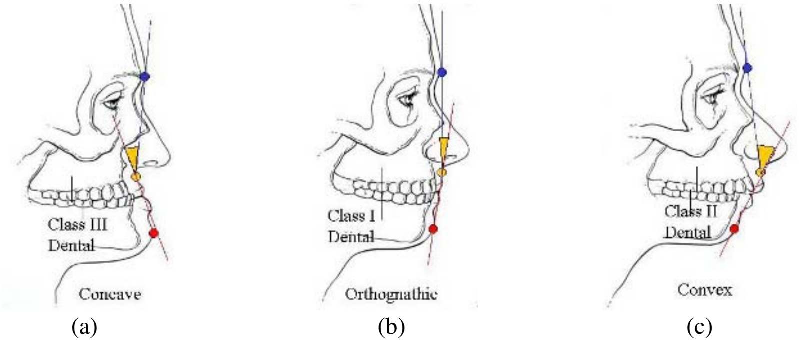
Koch et al. [10] developed a surgical planning and prediction system of human facial shape after maxillofacial surgery. After initial preprocessing, reconstruction, and registration, a finite element model of the facial surface and soft tissue is provided which is based on triangular finite elements. The resulting shape is generated from minimizing the global energy of the surface under the presence of external forces. Roth et al. [11] improved a finite element approach for volumetric soft tissue modeling in the context of facial surgery simulation. They propose an extension of linear elasticity towards incompressibility and nonlinear material behavior, in order to describe the complex properties of human soft tissue more accurately.

The existing research on the analysis of soft tissue changes after surgery in medicine suggests that this topic has great importance to surgical planning and treatment. Previous studies were limited to the 2D cephalometric device and other technical limitations. However, current approaches of the 3D analysis of soft tissue changes in clinical medicine still need more progress in computational accuracy and efficiency. In this study, we proposed a new approach of registering inter-patient CT scans using surface registration technique. Using a chamfer distance map, a registration speed is accelerated by referring to pre-calculated distance value without point-to-point correspondence identification. Our adaptive optimization approach dramatically reduces the registration time and improves the registration accuracy. We could validate this method to investigate the 3D changes of bone and soft tissue after mandibular setback surgery.

The organization of the paper is as follows. In Section 2, we introduce mandibular setback surgery. In section 3, we discuss how to segment the skull. We propose a surface registration method based on a chamfer distance map with adaptive optimization. In Section 4, a grid measurement is explained for detecting the changes of bone, and soft tissue. In Section 5, experimental results show that our method accurately and rapidly aligned the skull and the changes of bone, and soft tissue were measured. This paper is concluded with brief discussion of the results in Section 6.

2 Mandibular Setback Surgery

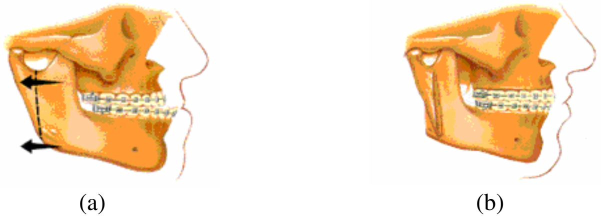
Orthognathic surgery involves the surgical manipulation of the elements of the facial skeleton to restore the proper anatomic and functional relationship in patients with dentofacial skeletal anomalies. Orthognathic surgery can be used to manage a broad spectrum of maxillofacial abnormalities [12]. Excess facial convexity, flatness, or concavity is felt to be less than ideal in Fig. 1.



(a) (b) (c)

Fig. 1. Profile analysis to classify a patient as (a) class III (b) class I (c) class II

Many orthodontic patients in class III have been reported to be severe enough to benefit from mandibular setback surgery [13]. Mandibular setback surgery can improve the occlusion, masticatory function, and aesthetics by changing the position of the mandible in Fig. 2. Various osteotomies are used to correct mandibular deformities, and the choice depends on the particular deformity. The sagittal split ramal osteotomy is the primary choice for correcting mandibular prognathism.



(a) (b)

Fig. 2. The procedure of mandibular setback surgery (a) Before surgery (b) After surgery

3 Automatic Skull Segmentation and Registration

Fig. 3 shows the pipeline of our method for skull segmentation and registration in pre- and post-operative CT scans. Since our method is applied to mandibular setback surgery, we can assume that the shape of upper skull in each CT scan is unchanged. Based on this assumption, we found that rigid transformation of upper skull surface would be adequate for the registration of pre- and post-operative CT scans.

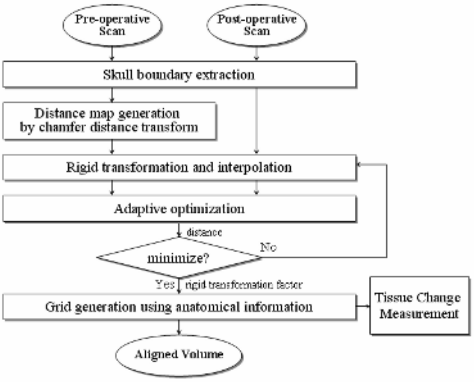
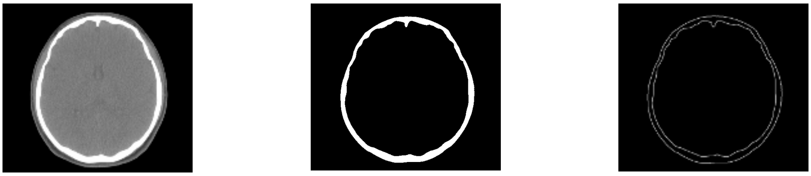

Pre-operative Scan
Post-operative Scan
Skull boundary extraction
Distance map generation by chamfer distance transform
Rigid transformation and interpolation
Adaptive optimization
distance
minimize?
No
Yes
rigid transformation factor
Grid generation using anatomical information
Tissue Change Measurement
Aligned Volume

Fig. 3. The pipeline of proposed method

3.1 Automatic Skull Segmentation

In this section, we describe an automatic segmentation method for identifying skulls. Our method consists of two steps: 1) the thresholding step to identify the region of skull, 2) the extraction step to delineate the skull edge (Fig. 4). In the thresholding step, skulls are separated from the surrounding anatomy by identifying pixels of skull based on the bone density value, which is larger than 150 HU(Housefield Unit). For each pixel, the pixel intensity is compared with the lower and upper thresholds. If the pixel value is inside the threshold range, the output pixel is assigned 255. Otherwise the output pixels are assigned 0. In the extraction step, image analysis to determine skull contours was performed by calculating the magnitude of the image gradient, which is computed using a simple finite difference approach. The image is convolved with masks of (-1, 0, 1) in x, y dimension, then adding the sum of their squares and computing the square root of the sum.



(a) (b) (c)

Fig. 4. The process of skull segmentation (a) original image (b) threshold image (c) edge image

3.2 Chamfer Distance Map Generation

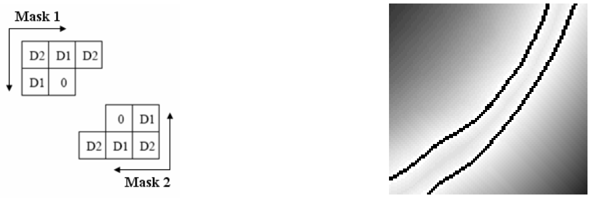
Chamfer distance transform [14] reduces the generation time of the distance map by an approximated distance transform compared to a Euclidean distance transform. Chamfer distance transform can be generated by performing a sequence of local operations while scanning image twice. Although our distance map is generated in 3D coordinate, we explain chamfer distance transform in 2D coordinate for an illustration. In forward scan, we compute $f_1(p)$ for all p∈ image in a single standard scan of image. For each p, f_1 has already been computed for all of the *q*s in B(p). If p has coordinates (x, y), B(p) contains (x, y+1), (x-1, y), (x-1,y+1) and (x+1, y+1).

$$f_1(p) = \begin{cases} 0 & \text{if p} \in \text{boundary} \\ \min\{f_1(q) + 1 : q \in B(p)\} & \text{if p} \notin \text{boundary} \end{cases} . \tag{1}$$

In backward scan, we compute $f_2(p)$ for all p∈ image in a single reverse standard (right-to-left, bottom-to-top) scan of image. A(p) contains the remaining neighbors of p, which are not contained in B(p).

$$f_2(p) = \min\{f_1(p), f_2(q) + 1 : q \in A(p)\} . \tag{2}$$

The computation of distance is performed by the two-step distance transformation of forward and backward masks, which implements above algorithm efficiently in Fig. 5(a). We implement chess-board distance transform. Fig. 5(b) shows the result of the chamfer distance map in which darker pixel has larger distance from the boundary.



(a) (b)

Fig. 5. Distance map (a) forward and backward masks (b) the result of chamfer distance map

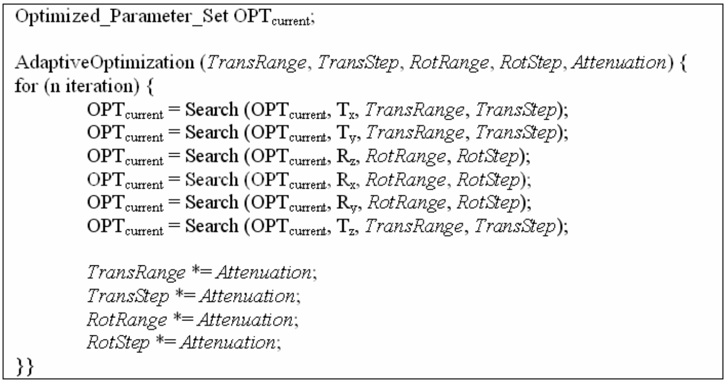
3.3 Surface Registration Using Adaptive Optimization

The distance measure in Eq. (3) is employed to determine the degree of resemblance of skull surfaces of pre- and post-operative volume. The average surface distance between two surfaces(*ASD*) reaches the minimum when skull boundary points of pre- and post-operative volumes are accurately matched.

$$ASD = \frac{1}{N_{post}} \sum_{i=0}^{N_{post}-1} \text{DistanceMap}_{\text{pre}}(Trasform(P_{post}(i))) , \tag{3}$$

where $\text{DistanceMap}_{\text{pre}}(P)$ is the distance value of P in the 3D distance map of pre-operative volume. $Trasform(P)$ is the rigid transformation of the point P in post-operative volume. $P_{post}(i)$ is ith boundary point of post-operative volume. N_{post} is the total number of surface points in post-operative volume.

Powell's direction set method in multidimensions is then used to minimize *ASD* using Brent's one-dimensional optimization algorithm [15]. We propose the adaptive optimization technique to change the search space and step dynamically to improve computational efficiency and robustness of Powell's direction set method. In Fig. 6, the procedure of our adaptive optimization technique is described as pseudo code. Due to rigid transformation, search parameters are limited to translational, and rotational values. At the first iteration, the search range is wide and the search step is coarse. At the next iteration, the search range becomes narrower and the search step becomes finer as the factor of *Attenuation*. With this approach, the search space can be extended for robust optimization without sacrificing computational efficiency.



```
Optimized_Parameter_Set OPT_current;

AdaptiveOptimization (TransRange, TransStep, RotRange, RotStep, Attenuation) {
for (n iteration) {
        OPT_current = Search (OPT_current, T_x, TransRange, TransStep);
        OPT_current = Search (OPT_current, T_y, TransRange, TransStep);
        OPT_current = Search (OPT_current, R_z, RotRange, RotStep);
        OPT_current = Search (OPT_current, R_x, RotRange, RotStep);
        OPT_current = Search (OPT_current, R_y, RotRange, RotStep);
        OPT_current = Search (OPT_current, T_z, TransRange, TransStep);

        TransRange *= Attenuation;
        TransStep *= Attenuation;
        RotRange *= Attenuation;
        RotStep *= Attenuation;
}}
```

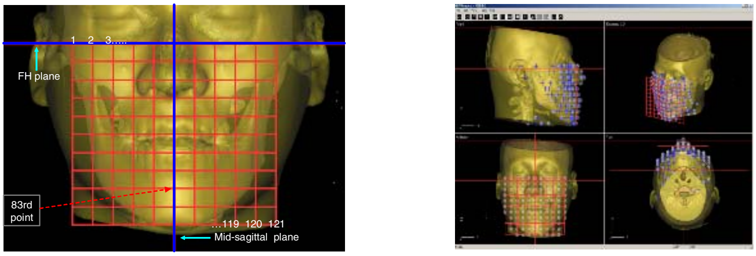
Fig. 6. Pseudo code of adaptive optimization

4 Grid Generation for Tissue Change Measurement

To analyze surgical changes, the grid, defined by the cephalometric landmarks, was formed parallel to the coronal plane. When a ray is orthogonally projected to the coronal plane, the intersection points of soft and hard tissue was assumed to represent the corresponding soft and hard tissue points of each patient. The grid definition is as follows; the upper border of the grid was FH plane, lower border was Me, left border was left Po, and right border was right Po. All the cephalometric landmarks to define the grid were summarized in Table 1. The grid was also created perpendicular to FH plane, in front of the surface model of the soft tissue part in Fig. 7. Then, the length from Me to FH plane was divided into 10 equal parts. Finally 11 horizontal lines were generated including the upper and lower border lines. The length from left Po to the mid-sagittal plane and the length from right Po to the mid-sagittal plane were compared. The shorter length was chosen and evenly divided into 5 parts. Mirroring those points on the basis of the mid-sagittal plane, new corresponding points were generated. Similarly 11 vertical lines were also created. A total of 121 points for measuring the surgical changes were defined by the intersection of these lines.

Table 1. Landmarks used for the grid definition

Landmark	Definition
Polt	The highest point on the upper margin of the left external auditory meatus
Port	The highest point on the upper margin of the right external auditory meatus
Me	The most inferior point on the symphysis of the mandible in the medial plane
Orlt	The lowest point on the lower margin of left orbit
Orrt	The lowest point on the lower margin of right orbit
FH plane	The plane defined by Polt, Port, Orlt, Orrt



(a) (b)

Fig. 7. Measurement using grid (a) grid generation (b) point projection

All points were projected onto the coronal plane through the skull and soft tissue. The coordinates of all the intersected points on the skull and the soft tissue from the projected ray were calculated. If there was no crossing with the skull or soft tissue, the point was regarded as missing. X axis was defined in the left-right direction, y axis in the antero-posterior direction, and z axis in the caudal-cephalic direction. The y axis value was analyzed for the antero-posterior changes after surgery.

5 Experimental Results

All our implementation and test were performed on an Intel Pentium IV PC containing 3.4 GHz and 2.0 GB of main memory. Our method has been applied to eight clinical datasets with mandibular prognathism, as described in Table 2.

Table 2. Image conditions of experimental datasets

Subject #		Volume size	Pixel size (mm)	Slice spacing (mm)	Subject #		Volume size	Pixel size (mm)	Slice spacing (mm)
1	Pre-operative	512x512x256	0.42x0.42	1.0	5	Pre-operative	512x512x241	0.48x0.48	1.0
	Post-operative	512x512x375	0.39x0.39	0.6		Post-operative	512x512x197	0.39x0.39	1.3
2	Pre-operative	512x512x237	0.47x0.47	1.0	6	Pre-operative	512x512x282	0.49x0.49	1.0
	Post-operative	512x512x245	0.44x0.44	1.0		Post-operative	512x512x273	0.45x0.45	1.0

Table 2. (*continued*)

3	Pre-operative	512x512x272	0.46x0.46	1.0	7	Pre-operative	512x512x281	0.46x0.46	1.0
	Post-operative	512x512x253	0.45x0.45	1.0		Post-operative	512x512x277	0.46x0.46	1.0
4	Pre-operative	512x512x264	0.42x0.42	1.0	8	Pre-operative	512x512x157	0.45x0.45	1.6
	Post-operative	512x512x237	0.41x0.41	1.0		Post-operative	512x512x240	0.46x0.46	1.0

Fig. 8(a), (b) shows 3D volume rendering of the patient with mandibular prognathism before and after mandibular setback surgery. We can recognize that the protrusion of the lower jaw in Fig. 8(a) is relaxed by changing the position of the mandible, which improves the occlusion, masticatory function, and aesthetics. Fig. 8(c), (d) shows the effectiveness of our surface registration. The transitional and rotational difference between pre- and post-operative volume shown in Fig. 8(c) is much reduced by our method as shown in Fig. 8(d).

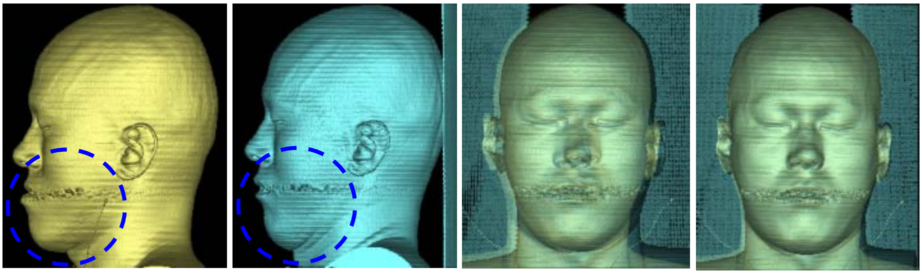

(a) (b) (c) (d)

Fig. 8. The results of registration (a) pre-operative volume (b) post-operative volume (c) before registration (d) after registration

Fig. 9 shows the registration results of our method in comparison with the conventional method using no additional technique. The average ASD of the conventional method and our method for eight patients is 2.67, 2.10(voxel), respectively while the average of ASD before registration is 6.93. With our adaptive optimization approach, the registration accuracy is much improved.

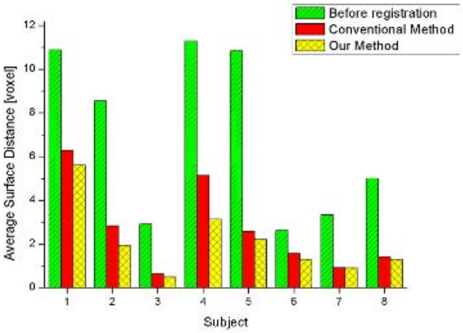


Fig. 9. Comparison of the registration accuracy using ASD

Fig. 10 shows the registration time of our method in comparison with the conventional method. The average registration time of the conventional method, and our method for eight patients is 14, and 9(sec), respectively. Our adaptive optimization approach reduces the registration time much. The average of total processing time of our method for eight cases is summarized in Table 3. For eight subjects, it takes less than 1 minute.

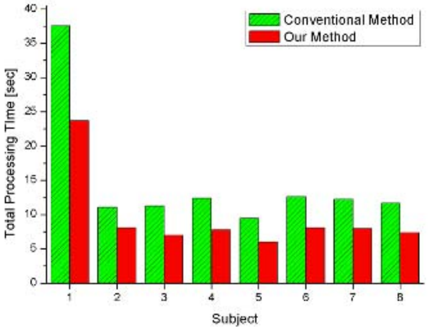


Fig. 10. Comparison of the surface registration time

Table 3. Average of total processing time for 8 cases (sec)

Distance map generation	Registration	Total processing
28	9	37

The changes of bone and soft tissue were measured in Table 4. The manual method is registered by the orthodontist using the cephalometric landmarks based on the anatomical knowledge of anthropometry. To compare the performance of our method, the absolute error of anterior-posterior (y coordinate) of corresponding points of the grid between pre- and post-operative was analyzed by using the standard paired t-test and its nonparametric counterpart the Wilcoxon signed-rank test. Nonparametric tests are more robust for small samples (in this case n = 8) and do not rely on assumptions of normality for the underlying distributions. The statistical test shows that there is significant difference between two methods ($p < 0.05$).

Table 4. Comparison of manual registration and our automatic registration method using the difference of anterior-posterior (y coordinate) of corresponding points of the grid

	Manual registration		Our method	
	Bony movement (Mean ± S.D.)	Soft tissue movement (Mean ± S.D.)	Bony movement (Mean ± S.D.)	Soft tissue movement (Mean ± S.D.)
Absolute error	4.05 ± 4.03	5.87 ± 6.14	2.95 ± 5.03	3.46 ± 2.29

6 Conclusion

We have developed a new automated system that registers pre- and post-operative dental CT scans for a three-dimensional analysis on soft and hard tissue changes after mandibular setback surgery. Our method matches automatically extracted skull to obtain optimal registration parameters. Using chamfer distance map, a registration speed is accelerated by referring to pre-calculated distance value. Our adaptive optimization approach reduces the registration time and improves the registration accuracy. Eight pairs of pre- and post-operative CT scans have been used for the performance evaluation. Experimental results show that our algorithm is more accurate, and converges faster than conventional ones. All our registration process is finished within 1 minute. Using a grid measurement, the changes of bone, and soft tissue were measured in skeletal Class III mandibular prognathism patients. Our method could be applicable to other oral and maxillofacial surgeries as well as plastic surgeries.

References

1. Hershey, H.G., Smith, L.H., Soft-tissue Profile Change Associated with Surgical Correction of the Prognathic Mandible, Am. J. Orthod, Vol. 65 (1974) 483-502.
2. Bjőrk, A., The Relationship of the Jaws to the Jaws to the Cranium, Introduction to orthodontics, McGraw Hill Book Company (1960) 104-140.
3. Bjőrk, A., Skiller, V., Normal and Abnormal Growth of the Mandible. A Synthesis of Longitudinal Cephalometric Implant Studies over a Period of 25 Years, Eur. J. Orthod, Vol. 5 (1983) 1-46.
4. Brodie, A.G., Late Growth Changes in the Human Face, Angle Orthod, Vol. 23 (1953) 146-157.
5. Steiner, C.C., Cephalometrics for You and Me, Am. J. Orthod, Vol. 39 (1953) 729-755.
6. McNeill, R.W., Proffit, W.R., White, R.P., Cephalometric Prediction for Orthodontic Surgery, Angle Orthod, Vol. 42 (1972) 154-164.
7. McCance, A.M., Moss, J.P., Wright, W.R., Linney, A.D., James, D.R., A Three Dimensional Soft Tissue Analysis of 16 Skeletal Class III Patients Following Bimaxillary Surgery, Br. J.Oral Maxillofac. Surg., Vol. 30 (1992) 221-232.
8. Moss, J.P., Ismail, S.F., Hennessy, R.J., Three-dimensional Assessment of Treatment Outcomes on the Face, Orthod. Craniofac. Res., Vol. 6 (2003) 126-131.
9. McCance, A.M., Moss, J.P., Fright, W.R., James, D.R., Linney, A.D., A Three Dimensional Analysis of Soft and Hard Tissue Changes Following Bimaxillary Orthognathic Surgery in Skeletal III Patients, Br. J. Oral Maxillofac. Surg., Vol. 30 (1992) 305-312.
10. Koch, R.M., Gross, M.H., Carls, F.R., Büren, D.F., Parish, Y.I.H., Simulating Facial Surgery Using Finite Element Models, Proceedings of ACM SIGGRAPH (1996) 421-428.
11. Roth, S.H.M., Gross, M.H., Turello, S., Carls, F.R., A Bernstein-Bézier Based Approach to Soft Tissue Simulation, Compuber Graphics Forum, Vol. 17, No. 3 (1998) 285-294.
12. Patel, P.K., Han, H., Kang N.H., Craniofacial, Orthognathic Surgery, emedicine (2004).
13. Chen, F., Terada, K., Hanada, K., Saito, I., Predicting the Pharyngeal Airway Space After Mandibular Setback Surgery, Journal of Oral and Maxillofacial Surgery, Vol. 63, No. 10, (2005) 1509-1514
14. Butt, M.A., Maragos, P., Optimum Design of Chamfer Distance Transforms, IEEE Transactions on Image Processing, Vol. 7, No. 10 (1998) 1477-1484.
15. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., Numerical Recipes in C – the Art of Scientific Computing, Cambridge University Press (1992).

Voting Method for Stable Range Optical Flow Computation

Atsushi Imiya[1] and Daisuke Yamada[2]

[1] Insutitute of Media and Information Technology, Chiba University, Japan
[2] School of Science and Technology, Chiba University, Japan

Abstract. For the non-invasive imaging of moving organs, in this paper, we develop statistically accurate methods for the computation of optical flow. We formalise the linear flow field detection as a model-fitting problem which is solved by the least squares method. Then, we show random-ssampling-and-voting method for the computation of optical flow as model-fitting problem. We show some numerical examples which shows the performance of our method.

1 Introduction

Optical flow is a non-invasive and non-interactive technique for the detection of motion of an object. Therefore, for medical study and diagnosis of moving organs in human body, optical flow of tomographic images provides a fundamental tool [1]. The non-invasive imaging of moving organs is achieved by NMR, X-ray, and ultrasonic. Usually the signal-to-noise ratio of non-invasive imaging is low. Therefore, we are required to develop statistically accurate methods for the computation of optical flow for tomographic images.

In this paper, we deal with the random sampling and voting process for linear flow detection. The method is an extension of the randomised Hough transform which was first introduced in [5] for planar image analysis. Later they applied the method to planar motion analysis [3] and shape reconstruction from flow field detection [4]. These results indicates that the inference of parameters by voting solves the least-squares problem in machine vision without assuming the predetermination of point correspondences between image frames. We show that the randomised sampling and voting process detects optical flow.

The slope selection problem in computational geometry [7] finds a pair of sample points on a plane which defines a line to approximate a distribution of sample points. Theil-Sin estimator selects a pair of points which yield the medians of two parameters of lines [6]. The selection process of pairs of points from samples which derive a line has the same mathematical structure with the randomised Hough transform [6,3,4,5]. This process is valid if the number of sample points is sufficiently large. Combining the ideas of Theil-Sen estimator and the theory of generalised inverse of matrix, we propose a method to estimate robustly the solution of the least mean squares problem for optical flow detection. The classical Hough transform estimates the parameters of models. In the classical

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 332–341, 2006.

Hough transformation, the accumulator space is prepared for the accumulation of the voting for the detection of peaks which correspond to the parameters of models to be detected. In this paper, we investigate for the data mining in the accumulator space for the voting method, which is a generalisation of the Hough transform, since the peak detection in the Hough transform could be considered as the data discovery in the accumulator space.

2 Geometry for Linear Optimization and Estimation

Many problems in computer vision are expressed as the minimization of the criterion

$$J(\boldsymbol{u}) = |\boldsymbol{A}\boldsymbol{u} - \boldsymbol{d}|^2 \tag{1}$$

for an $n \times m$ matrix $\boldsymbol{A}$ and an n-dimensional vector $\boldsymbol{b}$. This minimization problem is also described in the form

$$K(\boldsymbol{v}) = |\boldsymbol{F}\boldsymbol{v}|^2,\ \boldsymbol{F} = (\boldsymbol{A}, -\boldsymbol{d}),\ \boldsymbol{v} = \begin{pmatrix} \boldsymbol{u} \\ 1 \end{pmatrix}, \boldsymbol{F} = \begin{pmatrix} \boldsymbol{f}_1^\top \\ \boldsymbol{f}_2^\top \\ \vdots \\ \boldsymbol{f}_n^\top \end{pmatrix} \tag{2}$$

since

$$\boldsymbol{A}\boldsymbol{u} - \boldsymbol{d} = 0 \Leftrightarrow \boldsymbol{F}\boldsymbol{v} = 0, \tag{3}$$

For the first expression in eq. (1), the LMS solution is given as $\boldsymbol{u} = \boldsymbol{A}^\dagger \boldsymbol{c}$, where $\boldsymbol{A}^\dagger$ is the Moore-Penrose inverse of matrix $\boldsymbol{A}$. For $n \times m$ matrix $\boldsymbol{A}$, if the rank of $\boldsymbol{A}$ is m, $\boldsymbol{A}^\dagger = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top$.

Equation (3) means that $\boldsymbol{v} \in N(\boldsymbol{F})$ for $\boldsymbol{v} = (\boldsymbol{u}^\top, 1)$ is the solution of the minimization problem. Since the elements of matrix $\boldsymbol{F}$ are usually depending only data, we are required to design a robust method to solve this linear system of equations.

Since

$$N(\boldsymbol{F}) = R(\boldsymbol{F}^\top)^\perp = R(\boldsymbol{F}\boldsymbol{F}^\top)^\perp, \tag{4}$$

the solution is the eigenvector associated with the minimum eigenvalue of a $(m+1) \times (m+1)$ matrix $\boldsymbol{M} = \boldsymbol{F}\boldsymbol{F}^\top$. If the rank of $\boldsymbol{F}$ is m, the minimum eigenvalue is zero. Therefore, the solution which minimizes the criterion $K(\boldsymbol{v})$ lies in the null space of matrix $\boldsymbol{F}$.

Assume that our problem is to estimate a m dimensional-vector $\boldsymbol{u}$ from a system of equations, $\boldsymbol{f}_\alpha^\top \boldsymbol{v} = 0$. Each equation of this system of equations is considered to be a constraint in a minimization problem of a model-fitting process. Since each constraint determines a hyperplane in the m-dimensional Euclidean space, the common point of a pair of equations,

$$\{\boldsymbol{u}_S\} = \bigcap_{\alpha \in S} \{\boldsymbol{u} | \boldsymbol{f}_\alpha^\top \boldsymbol{v} = 0, \boldsymbol{v} = (\boldsymbol{u}^\top, 1)^\top\}, \tag{5}$$

for S is a subset of $1 \le s \le n$ such that $|S| = m$, where $|A|$ is the number of the elements of set A, an estimator of the solution which satisfies a collection of constraints. Since we have n constrains, we can have $\begin{pmatrix} n \\ m \end{pmatrix}$, estimators as the common points of the collection of linear constraints.

3 Range Flow Detection

3.1 Range Images

We define the geometry for the detection of range images. Setting $(x, y, z)^\top$ to be the world cordinate in a work space, we assume that our range data of object measured as the depth $-z$ at the point $(x, y, 0)^\top$ on the plane $z = 0$. Therefore, the depth of object from imaging plane $z = 0$ is expresed as

$$-z = f(x, y). \tag{6}$$

Setting

$$g(x, y, z) = f(x, y) + z, \tag{7}$$

the level $g(x, y, z) = 0$ expresses the range data in the z direction. Therefore, a spatial image $g(x, y, z) = f(x, y) + z$ defines a range image.

3.2 Optical Flow of Spatial Images

In three-dimensional Euclidan space $\mathbf{R}^3$, the total derivative of the temporal function $g(x, y, z, t)$ with respect to time argument t is

$$\frac{d}{dt}g = g_x\dot{x} + g_y\dot{y} + g_z\dot{z} + g_t. \tag{8}$$

Assuming $\frac{d}{dt}g = 0$, $(u, v, w)^\top = (\dot{x}, \dot{y}, \dot{z})^\top$ is optical flow which expresses the motion of each point. For a sequence of temporal range-images, optical flow is the solution of

$$f_x\dot{x} + f_y\dot{y} + \dot{z} + f_t = 0, \tag{9}$$

if $g(x, y, z, t) = f(x, y, t) + z$.

Assuming that flow vector $\boldsymbol{u} = (u, v, w)^\top$ is constant in an area Ω, whose centre is at $\boldsymbol{x} = (x, y)$, optical flow for time $t = \tau$ at point $\boldsymbol{x} = (x, y, z)^\top$, is the solution of a system of equations

$$a_\alpha u + b_\alpha v + c_\alpha w + d_\alpha = 0,\ \alpha = 1, 2, \cdots, n,\ n \ge 2 \tag{10}$$

where

$$\begin{aligned} a_\alpha &= g_x(x, y, z, t)|_{x=x_\alpha, y=y_\alpha, z=z_\alpha, t=\tau}, \\ b_\alpha &= g_y(x, y, z, t)|_{x=x_\alpha, y=y_\alpha, z=z_\alpha, t=\tau}, \\ c_\alpha &= g_z(x, y, z, t)|_{x=x_\alpha, y=y_\alpha, z=z_\alpha, t=\tau}, \\ d_\alpha &= g_t(x, y, z, t)|_{x=x_\alpha, y=y_\alpha, z=z_\alpha, t=\tau}, \end{aligned}$$

and $\boldsymbol{x}_\alpha = (x_\alpha, y_\alpha, z_\alpha)^\top$ is a point in the windowed area Ω.

3.3 Flow Computation by Random Sampling

Next, we propose a simple and effective method for solving the system of linear equations defined by eqs. (10) and (11). Our problem is to estimate a three-dimensional vector $\boldsymbol{u} = (u, v, w)^\top$ from a linear equation $\boldsymbol{A}\boldsymbol{u} + \boldsymbol{d} = 0$ for

$$\boldsymbol{A} = \begin{pmatrix} a_1, & b_1, & c_1 \\ a_2, & b_2, & c_2 \\ \vdots & \vdots & \vdots \\ a_n, & b_n, & c_n \end{pmatrix}, \boldsymbol{u} = \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \boldsymbol{d} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix}, \tag{11}$$

when the rank of matrix $\boldsymbol{A}$ is three. Each equation $a_\alpha u + b_\beta v + c_\alpha w + d_\alpha = 0$ is considered to be a constraint. Since each constraint determines a plane on the u-v-w space, the common point of a triplet of equations,

$$\begin{aligned} \{\boldsymbol{u}_{ijk} = (u_{ijk}, v_{ijk}, w_{ijk})^\top\} = &\{(u, v, w)^\top | a_i u + b_i v + c_i w + d_i = 0\} \\ &\cap\{(u, v, w)^\top | a_j u + b_j v + c_j w + d_j = 0\} \\ &\cap\{(u, v, w)^\top | a_k u + b_k v + c_k w + d_k = 0\} \end{aligned} \tag{12}$$

for $i \neq j \neq k \neq i$, is an estimator of the solution which satisfies a collection of constraints.

From a triplet of system of equations, $a_\alpha u + b_\alpha v + c_\alpha w + d_\alpha = 0$, for $\alpha \in \{i, j, k\}$, we compute vector $\boldsymbol{a} = (A, B, C, D)^\top$, where

$$A = \begin{vmatrix} b_i & c_i & d_i \\ b_j & c_j & d_j \\ b_k & c_k & d_k \end{vmatrix} \quad B = -\begin{vmatrix} a_i & c_i & d_i \\ a_j & c_j & d_j \\ a_k & c_k & d_k \end{vmatrix}$$

$$C = \begin{vmatrix} a_i & b_i & d_i \\ a_j & b_j & d_j \\ a_k & b_k & d_k \end{vmatrix} \quad D = -\begin{vmatrix} a_i & b_i & c_i \\ a_j & b_j & c_j \\ a_k & b_k & c_k \end{vmatrix}.$$

If and only if $D \neq 0$, we obtain $\boldsymbol{u} = (\frac{A}{D}, \frac{B}{D}, \frac{C}{D})^\top$.

For matrix $\boldsymbol{D}$, such that,

$$\boldsymbol{D} = \begin{pmatrix} a_i & b_i & c_i \\ a_j & b_j & c_j \\ a_k & b_k & c_k \end{pmatrix}, \tag{13}$$

the matrix $\boldsymbol{D}^\top\boldsymbol{D}$ is an approximation of the structure tensor

$$\boldsymbol{S}_\Omega = \int\int\int_\Omega \boldsymbol{S} dxdydz, \ \boldsymbol{S} = \begin{pmatrix} g_x^2 & g_x g_y & g_x g_z \\ g_x g_y & g_y^2 & g_y g_z \\ g_x g_z & g_y g_z & g_z^2 \end{pmatrix}, \tag{14}$$

since for $\boldsymbol{S}_\alpha = \boldsymbol{S}|_{\boldsymbol{x}=(x_\alpha, y_\alpha, z_\alpha)^\top}$, we have the relation

$$\boldsymbol{D}^\top\boldsymbol{D} = \sum_{\alpha=i,j,k} \boldsymbol{S}_\alpha, \ \boldsymbol{x}_\alpha \in \Omega, \tag{15}$$

$$\boldsymbol{A}^\top\boldsymbol{A} = \sum_{\forall \boldsymbol{x}_\alpha \in \Omega} \boldsymbol{S}_\alpha. \tag{16}$$

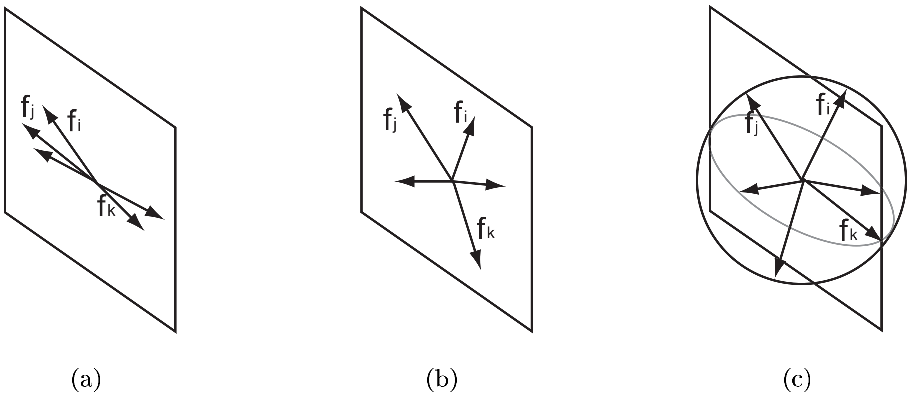


Fig. 1. Dimension Propeerty of Flow: Structure tensor defines local dimensionality of images

The rank of the structure tensor describes the the local structure of motion of object in an image. Equations (14) and (15) imply that the rank of matrix $\boldsymbol{D}$ is equivalent to the rank of sampled value of the structure tensor of spatial gradient.

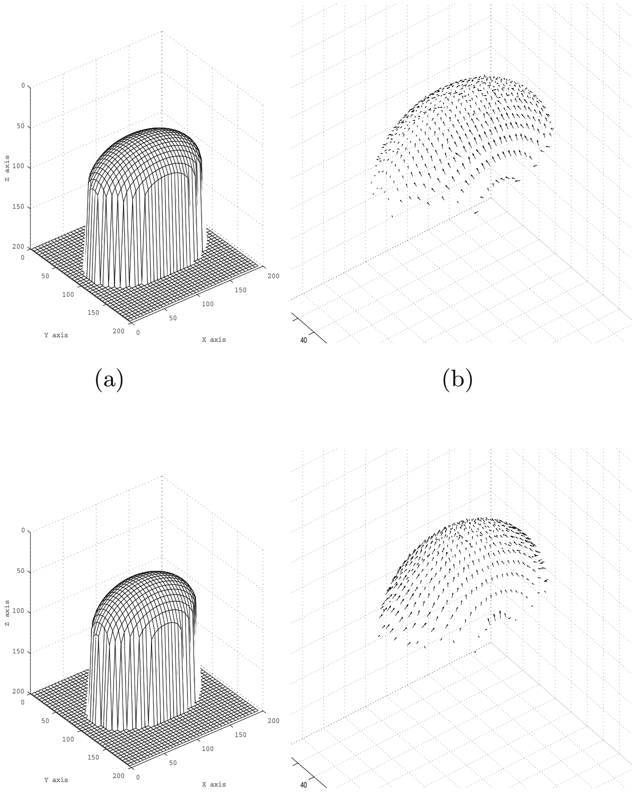
If and only if $D = 0$, $\boldsymbol{g}_\alpha = (a_\alpha, b_\alpha, c_\alpha)^\top$ at three deferent points $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, and $\boldsymbol{x}_k$ are independent. If and only if ranks of matrix $\boldsymbol{D}$ are 1 and 2, D is zero. If and only if rank of $\boldsymbol{D}$ is one, the spatial gradients at three points are parallel in region Ω. The is the three-dimensional analogous of the configuration of spatial gradients which causes the aperture problem for the planar problem.

If and only if rank $\boldsymbol{D}$ is two, the spatial gradient of a point lies on a plane spanned by spatial gradients of the other two points. This configuration means that the directional gradient in a direction is zero for all the points. For example, if the surface measured as range data and the iso-surface of a distribution is cylindrically symmetry for an axis, the rank of matrix $\boldsymbol{D}$ is two. For this configuration, the spatial gradients lie on a plane spanned by the eigenvectors $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ of $\boldsymbol{D}^\top \boldsymbol{D}$ corresponding to the non-zero eigenvalues. Therefore, there in no motion in the direction of eigenvector $\boldsymbol{h}_3$ of $\boldsymbol{D}^\top \boldsymbol{D}$ corresponding to the zero eigenvalue. On the slice perpendicular to $\boldsymbol{h}_3$, we can detect optical flow. These considerations clarify that our method automatically omits the aperture problem for spatial optical flow detection. Figure 1 shows local dimensionality of an image characterized by structure tensor.

Since we have n constraints and $\binom{n}{3} = \frac{n!}{3!(n-3)!}$, we can have $O(n^3)$ estimators as the common points of pairs of lines. Because of noise and computational errors, the solutions distribute in $(u, v, w)^\top$ plane. We vote the solutions to $(u, v, w)^\top$ and accumulate them for estimating the stable solution. Therefore, the estimation of solutions from pairs of equations is mathematically the same procedure as the Hough transform for the detection of lines on a plane from a collection of sample points. Therefore, to speed up the computation time, we can adopt a random sampling process for the selection of pairs of constraints. This procedure derives the same process with the randomised Hough transform such as

1. Randomly select a triplet of equations from the system of equations $\boldsymbol{F}\boldsymbol{v} = 0$.
2. Solve this system of linear equations, if a triplet of equations are independent, otherwise go back to step 1.
3. Vote this solution to (u, v, w) plane.
4. After an appropriate number of iterations from step 1 to step 3, detect peaks in (u, v, w) space.
5. Apply statistical analysis to the peaks for the detection of the accurate solution.

Since a triplet of points in a space determines a plane, for the estimation of a plane from scatted data with many outliers, Theil-Sen estimator computes the medians of two parameters computed triplet of sample points. Theil-Sen



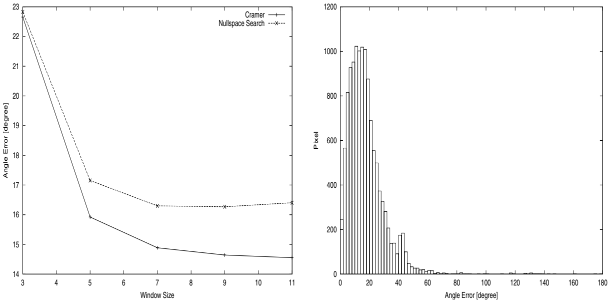
(a)

(b)

(c)

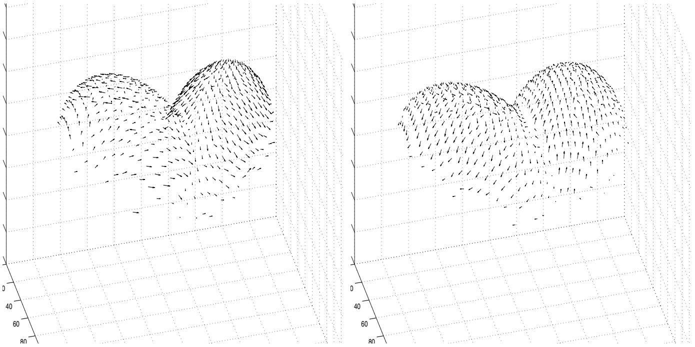
(d)

Fig. 2. A sequence of synthetic range images (a) and (c), and computed flow images (b) and (d)



(a) (b)

Fig. 3. (a) The angle errors against sizes of the windows 5 × 5, 7 × 7, 9 × 9, and 11 × 11. (b) The distribution of the angles between the theoretical and computed ones for pixels.



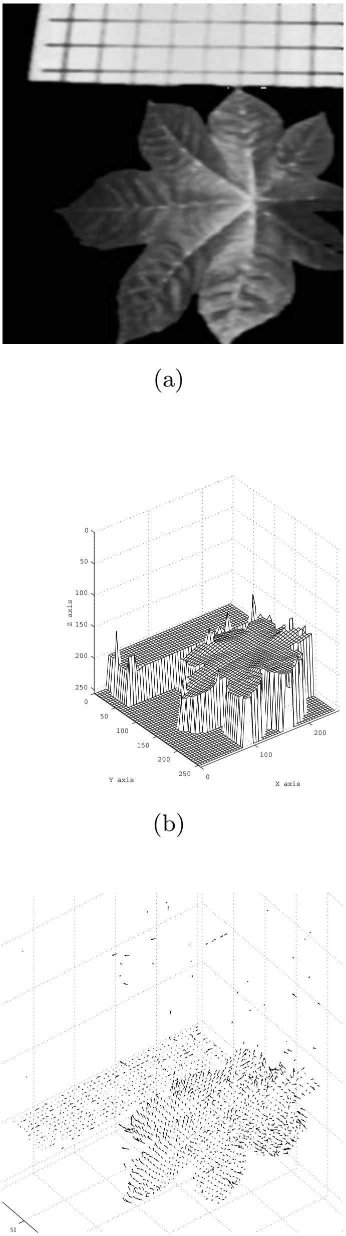
(a) (b)

Fig. 4. Flow of synthetic beating heart

estimator computes the medians of parameters which are computed from triplets of sample points from samples [6]. Estimation of the solution of linear system of equation using the common points of a collection of linear constraints has the same mathematical structure with Theil-Sen estimator. Therefore, for our problem setting $\boldsymbol{u}_{ijk} = (u_{ijk}, v_{ijk}, w_{ijk})^\top$, we adopt

$$\mathrm{median}\boldsymbol{u}_{ijk} = (\mathrm{median}u_{ijk}, \mathrm{median}v_{ijk}, \mathrm{median}w_{ijk})^\top \qquad (17)$$

as the solution from $\begin{pmatrix} n \\ 3 \end{pmatrix}$ solutions which are yield from collections of linear constraints. When $n \gg 3$, post-processing defined in eq. (17) becomes effective. We call this method the Matrix-Inversion Method.



(a)

(b)

(c)

Fig. 5. The result of Matric-Inversion Method for the practical data sequence "leaf" by H.Spies and J.Barron

4 Numerical Examples

For the detection of three-dimensional flow from sequences of range images, we evaluated the performance for the synthetic data. Our synthetic data is a moving ellipse $f(x, y, z, t) = 1$ such that

$$f(x, y, z, t) = \frac{x^2}{(a + \alpha \cos \omega t)^2} + \frac{y^2}{b + \beta \cos(\omega t + \frac{2}{3}\pi))^2} + \frac{z^2}{c + \gamma \cos(\omega t - \frac{2}{3}\pi))^2}$$

for $a = 70$, $b = 60$, $c = 50$, $\alpha = 10$, $\beta = 15$, $\gamma = 5$, $\omega = 2\pi/30$. In Figure 2, we show the image sequence for $t = 0, 5$.

We have evaluated our two methods. The first one directly computes the inverse of the 3×3 matrix with the Cramer method. Furthermore, the second method searches the vector in the null space using the singular value decomposition.

Figure 3 (a) shows graphs of the error-distribution of the two methods against the sizes of the windows. The errors of both methods degrease according to the sizes of the windows, since in a large window-area there exist many out-layers, both methods select accurate solution from many out-layers.

In Figure 3 (b), we show the distribution of the angles between the theoretical and computed ones for pixels. The average and variance of the angles are 17.2 [dig] and 12.6 [dig], respectively. Figure 3 (a) illustrates that the direct-matrix-inversion method is stable for the numerical computation. Figure 4 shows optical flow of the synthetic beating heart.

In Figure 5, we apply our method for the practical data sequence "leaf" by H.Spies and J.Barron. In Figure 5, (a) and (b) are the image and the range image of the same object. And (c) is the detected range flow. The result is compatible to the Horn-Schunck method for the range optical flow detection by Spies, Jahne, and Barron [2].

5 Conclusions

In this paper, we showed that the random sampling and voting process detects a linear flow field. We introduced a new method of solving the least-squares model-fitting problem using a mathematical property for the construction of a pseudo-inverse of a matrix. The greatest advantage of the proposed method is simplicity because we can use the same engine for solving multi-constraint problem with the Hough transform for the planar line detection.

A well studied method for the accurate optical flow computation is the application of multi-resolution analysis based on scale-space theory and pyramid transformation, because these method remove noise from images as preprocessing. Our method does not apply any preprocessing, though Theil-Sen estimator achieves a pose-processing for a collection of solutions.

Acknowledgement

We express our great thanks to Dr. H. Spies and Prof. J.Barron for arrowing us to use their range data set for numerical computation.

References

1. Song, S. M. and Leathy, R. M. Computation of 3-D verocity fields from 3-D cine CT images of a human heart, IEEE, Tr. Medical Imaging, **10**, 285-306, 1991.
2. Spies, H., Jahne, B., and Barron, J., Range Flow Estimation Computer Vision and Image Understanding, **85**, 209-231, 2002.
3. Kälviäinen, H., Oja, E., and Xu, L., Randomized Hough transform applied to translation and rotation motion analysis, *11th IAPR Proceedings of International Conference on Pattern Recognition*, 1992, 672-675.
4. Heikkonen, J., Recovering 3-D motion parameters from optical flow field using randomized Hough transform, *Pattern Recognition Letters*, **15**, 1995, 971-978.
5. Oja, E., Xu, L., and Kultanen, P., Curve detection by an extended self-organization map and related RHT method, Proc. International Neural Network Conference, **1**, 1990, 27-30.
6. Mount, D. M. and Netanyahu, N. S., Efficient randomized algorithms for robust estimation of circular arcs and aligned ellipses, Computational Geometry:Theory and Applications, **19**, 2001, 1-33.
7. Cole,R., Salowe, J. S., Steiger, W. L., and Szemeréd E., An optimal-time algorithm for slope selection, SIAM J. Computing, **18**, 1989, 792-810.

Shadow Removal for Foreground Segmentation

Kuo-Hua Lo, Mau-Tsuen Yang, and Rong-Yu Lin

Department of Computer Science & Information Engineering,
National Dong Hwa University, Taiwan
mtyang@mail.ndhu.edu.tw

Abstract. Removing shadows casted by moving foreground objects in a scene is a critical problem for many vision-based applications. We propose two algorithms that examine color/texture invariants, and exploit spatial-temporal consistency to detect shadows efficiently and reliably. The first algorithm assumes a static background model while the second algorithm addresses the perturbations of dynamic background in natural scenes. The experimental results show that the proposed methods can detect penumbra as well as umbra in different kinds of scenarios under various illumination conditions.

Keywords: foreground segmentation; shadow removal; photometric invariants; penumbra.

1 Introduction

Motion analysis in video sequence is important in many applications such as visual surveillance, obstacle tracking/recognition, video content analysis and Intelligent Transportation Systems (ITS). However, one of the main challenges is that moving cast shadows on the background could be classified as foreground objects by mistake. The performance of the successive analysis, recognition or tracking would be seriously degraded due to this problem. A reliable and efficient shadow removal algorithm is required before the potential power of these vision-based applications can be realized.

To distinguish shadow from foreground is quite difficult because both shadow and foreground look quite different from the background. Moreover, the cast shadow usually moves along the foreground object such that they share the same motion. In fact, we are only interested in the moving shadow since static shadow can be modeled as a part of the background. For this purpose, we defined two photometric invariants that are independent of the effect of shadow: the between-pixel invariants (texture feature) and within-pixel invariants (color feature). Two algorithms were proposed to remove shadow by utilizing these two kinds of invariants, neighborhoods and temporal consistency in the scenes. The first algorithm that assumes static background is efficient under stable scenes. The second algorithm that models dynamic background is more reliable in natural scenes including waving trees, rippling water, and rain/snow.

The remaining parts of this paper are organized as follows: Section 2 outlines and compares the related works. Section 3 shows the luminance model and defines the texture/color invariants. Section 4 presents the first shadow removal algorithm assuming static background. Section 5 describes the second shadow removal

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 342 – 352, 2006.

algorithm addressing dynamic background. Section 6 demonstrates the experimental results. Section 7 concludes this paper.

2 Related Works

Several shadow detection algorithms have been proposed for traffic surveillance. Generally speaking, shadow regions are detected based on information of the luminance, chrominance and gradient density. Large number of false alarms or miss detections can be reduced by the assumption of known geometry information of the foreground vehicles [2] or the lane lines [5]. As a result, these methods are only suitable for the road traffic applications. Wang et al. [9] proposed a shadow removal method that estimates the illumination direction and then recovers the foreground vehicles based on the information of both object edges and attributes of shadow. However, the estimation of the illumination direction and the shadow attributes is not very reliable under changing weather conditions.

It is possible to detect shadow by examining the color information of each pixel. Siala et al. [6] presented a moving shadow detection algorithm by training the shadow samples in RGB color space with Support Vector Domain Description (SVDD). A minimal radius hypersphere was found to represent all training samples of shadow. Then the algorithm can decide whether a pixel is shadow or not by checking if the color of the pixel falls in the hypersphere. A training process was performed by manually segmenting the shadow region in a bootstrap manner. The issue of dynamic illumination was not addressed. Cucchiara et al. [1] considered the color independence property in the HSV color space to detect shadow. It is observed that if a pixel is covered by shadow, the hue and saturation components of the pixel only change within a certain limit. However, the hue components on pixels with saturated or poor illumination are usually unstable.

A few shadow detection algorithms used gray scale images as inputs. Stauder et al. [7] relied on brightness, edge and shading information to detect moving cast shadows in a textured background. Xu et al. [10] assumed that shadow often appears around foreground object and tried to detect shadow by extracting moving edges. Morphological filters were used intensively. Without considering color information, problems could occur when both the foreground and background are uniform regions without much texture.

Toth et al. [8] proposed a shadow detection algorithm based on color and shading information. First, the color space of an input image was converted from RGB to LUV. The image was segmented to several regions based on color information using mean shift algorithm. It is observed that, for every pixel in a small neighborhood in the umbra of a shadow region, the intensity values with shadow divided by those without shadow should be a constant. This shading property was used to detect umbra region. A successive morphological filter was applied to remove the penumbra region. The edge/texture information was not considered for shadow discrimination.

Funt et al. [3] presented a method for object recognition under changeful illumination by checking the equality of intensity ratios of neighboring pixels. Heikkila et al. [4] proposed a dynamic background subtraction method using texture features of local binary patterns. Although shadow detection is not within their scope, we found that the concept of color constancy and the idea of dynamic weighting list can be extended to remove shadow. In this paper, we define the color and texture invariants, and propose two shadow removal algorithms that combine the invariants with spatial-temporal consistency to remove penumbra as well as umbra in a scene.

3 The Luminance Model and Invariants

Suppose *I(x,y)* is the intensity value of the pixel located at *(x,y)*, *E(x,y)* is the irradiance of the 3D point projecting to *(x,y)* and *ρ(x,y)* is the diffuse reflectance of the same 3D point. A simple luminance model assuming Lambertian reflectance can be defined as follows:

$$I(x,y)=E(x,y)\rho(x,y) \tag{1}$$

Two kinds of shadow can appear in an image: the penumbra and the umbra. Their irradiance can be modeled by the following equation:

$$E(x,y)=\begin{cases} C_a+C_p\cdot\cos\theta & \text{no shadow} \\ C_a+k(x,y)\cdot C_p\cdot\cos\theta & \text{penumbra} \\ C_a & \text{umbra} \end{cases} \tag{2}$$

where C_a is the radiance of ambient light, C_p is the radiance of a distant light source and θ is the angle between the direction of the distant light source and the surface normal vector of the 3D point projecting to *(x,y)*. The weighting factor *k(x,y)* represents the percentage of the receiving energy when the distant light source is partially occluded (penumbra). The value of *k(x,y)* ranges from 0 (umbra) to 1 (no shadow).

3.1 Between-Pixel Invariants

A shadow casted on a background pixel changes its brightness instead of its texture. Real foreground and shadow pixels can be segmented by examining the invariability of the texture or edge information. Assuming the 3D points projecting to neighboring pixels receive the same irradiance, i.e., *E(x,y)=E(x+1,y)*, the intensity ratio between a pixel *(x,y)*and its neighboring pixel *(x+1,y)* on an image *I* can be calculated as follows:

$$\frac{I(x,y)}{I(x+1,y)}=\frac{E(x,y)\rho(x,y)}{E(x+1,y)\rho(x+1,y)}=\frac{\rho(x,y)}{\rho(x+1,y)} \tag{3}$$

As long as a pixel is projected by the same 3D point, this ratio (the intensities between neighboring pixels) should be invariant to illumination changes. In other words, the ratio should be roughly fixed no matter whether it is covered by shadow or not. We called this shading property as *invariants between neighboring pixels.*

3.2 Within-Pixel Invariants

A shadow casted on a background pixel changes its brightness but not changes much its color. Comparing pixel-wise color information between current image and background image can help to detect cast shadow. When a textureless foreground object is in front of a textureless background, the most important clue for separating cast shadow from foreground object is the color information. Suppose the spectral intensities of a pixel *(x,y)* are represented by *R(x,y)*, *G(x,y)*, and *B(x,y)* in RGB space. The spectral ratio between *R(x,y)* and *B(x,y)*can be defined as follows:

$$\frac{R(x,y)}{B(x,y)} = \frac{E_r(x,y)\rho_r(x,y)}{E_b(x,y)\rho_b(x,y)} = E_c(x,y)\cdot\frac{\rho_r(x,y)}{\rho_b(x,y)} \tag{4}$$

Assuming the color of the illumination do not change by the effect of shadow, the ratio $E_r(x,y)/E_b(x,y)$ should be equal to a constant $E_c(x,y)$. Thus, the spectral ratio *R(x,y)/B(x,y)* is invariant to the magnitude of the illumination and roughly equal to a constant even if it is covered by shadow. Similarly, the spectral ratio *G(x,y)/B(x,y)* is invariant under shadow or different illumination conditions. This spectral property is called *invariants within pixels.*

4 The Shadow Removal Algorithm Based on Static Background

The goal of the first algorithm is to separate shadow pixels from real foreground pixels as a refinement on the outputs of static background subtraction. Fig. 1 depicts a flowchart of the proposed shadow removal algorithm based on a static background. As a preprocessing step, a statistical background subtraction was applied to generate the foreground mask region (FMR). A noise removal algorithm was performed to refine the FMR. Then the proposed shadow detection algorithm was applied to the FMR by considering three factors: the between-pixel invariants, the within-pixel invariants and the spatial-temporal consistency.

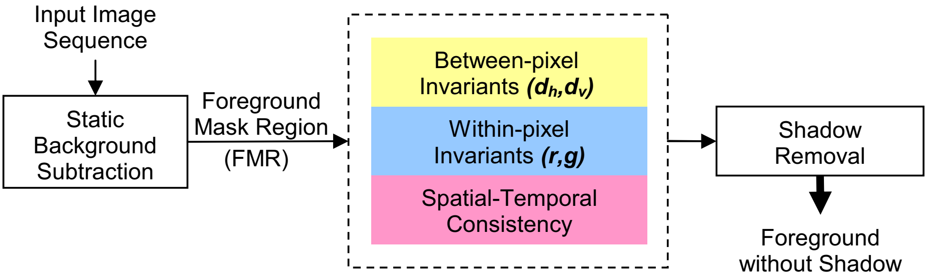

Input Image Sequence
Static Background Subtraction
Foreground Mask Region (FMR)
Between-pixel Invariants (d_h,d_v)
Within-pixel Invariants (r,g)
Spatial-Temporal Consistency
Shadow Removal
Foreground without Shadow

Fig. 1. Flowchart of the first shadow removal algorithm based on static background

Suppose *I* is the current image and *I'* is the background image. According to the between-pixel invariants discussed in section 3.1, the ratio of the intensities between neighboring shadow pixels in both current and background image should be the same, i.e.

$$\frac{I(x,y)}{I(x+1,y)} = \frac{I'(x,y)}{I'(x+1,y)} \quad \text{, if } (x,y) \text{ is in shadow region} \tag{5}$$

To speedup the examination of this property, two logarithm ratio maps, $d_h(x,y)$ and $d_v(x,y)$, for current image can be computed by convolving the logarithm image with a horizontal or vertical first-order derivative mask.

$$\begin{cases} d_h(x,y) = \ln \dfrac{I(x,y)}{I(x+1,y)} = \ln I(x,y) - \ln I(x+1,y) \\ d_v(x,y) = \ln \dfrac{I(x,y)}{I(x,y+1)} = \ln I(x,y) - \ln I(x,y+1) \end{cases} \tag{6}$$

The logarithm ratio maps *d_h'(x,y)* and *d_v'(x,y)* for background image can be defined similarly. A ratio map keeps the texture and edge information that should not be affected by cast shadow in an image. Based on this idea, a pixel is classified as shadow only if its value in the ratio map (texture information) is similar to those in the background. A simple pixel-wise comparison between *d(x,y)* and *d'(x,y)* can be used to determine whether a pixel belongs to shadow regions or not. Nevertheless, there could be some outliers due to noise or coincidence. To address this problem, spatial consistency is exploited to remove outliers. It is observed that shadows usually occupy a region instead of a few isolated pixels. The error score for discriminating the pixel *(x,y)* as shadow can be calculated by summing the difference of *d* and *d'* over all pixels in a small neighborhood window *W* centered at *(x,y)*:

$$\Psi(x,y) = \sum_{(i,j)\in W} \left| d_h(i,j) - d_h^{'}(i,j) \right| + \left| d_v(i,j) - d_v^{'}(i,j) \right| \tag{7}$$

By considering the spatial consistency in neighborhood, the overall error score $\Psi(x,y)$ for shadow discrimination is much more stable and outliers can be effectively reduced.

Suppose *(R, G, B)* represents the spectral information of a pixel in current image and *(R', G', B')* indicates the spectral information of the same pixel in background image. According to the within-pixel invariants defined in section 3.2, the spectral ratio in both current and background image should be the same, i.e.,

$$\frac{R(x,y)}{B(x,y)} = \frac{R'(x,y)}{B'(x,y)} \quad \text{, if } (x,y) \text{ is in shadow region} \tag{8}$$

For speedup purpose, two logarithm ratio maps, *r(x,y)* and *g(x,y)*, for current image can be computed by

$$\begin{cases} r(x,y) = \ln \dfrac{R(x,y)}{B(x,y)} = \ln R(x,y) - \ln B(x,y) \\ g(x,y) = \ln \dfrac{G(x,y)}{B(x,y)} = \ln G(x,y) - \ln B(x,y) \end{cases} \tag{9}$$

Since the value of *r* and *g* remain roughly the same under different illumination condition. The score of error for discriminating the pixel *(x,y)* as shadow is defined as:

$$\Theta(x,y) = \left| r(x,y) - r'(x,y) \right| + \left| g(x,y) - g'(x,y) \right| \tag{10}$$

where *r'* and *g'* are the spectral ratio of the background image. A smaller *Θ(x,y)* represents that the color of the pixel *(x,y)* does not change much and it is more likely to be a shadow pixel.

Methods considering only the between-pixel invariants (texture) can not distinguish between a foreground without texture and its shadow on a uniform background. Methods considering only the within-pixel invariants (color) tend to wrongly classify a foreground region with similar color as its background to be a shadow. Assuming the foreground object moves slowly, temporal consistency between frames can provide a clue for potential shadow regions. In other words, if the frame rate is high, a shadow pixel at time instant *t* tends to remain in shadow region at time *t+1*. Exploiting temporal consistency can prevent wrongly classifying the temporally isolated noise as shadow regions. A reliable shadow detection system should be able to consider all these factors simultaneously. In our system, the error scores corresponding to these factors are fused together using the following recursive linear equation:

$$\Omega_t(x,y) = \frac{a \cdot \Psi_t(x,y) + b \cdot \Theta_t(x,y) + c \cdot \Omega_{t-1}(x,y)}{a+b+c} \tag{11}$$

where *a*, *b* and *c* are weighting parameters that control the importance of each factor and the speed of the recursive update. Their values are determined empirically in our current experiments and can be adjusted dynamically for better adaptability in the future. For example, the weight *a* should be lowered for images without much texture; the weight *b* should be lowered for images under saturated or poor illumination; the weight *c* should be lowered for images with fast moving objects. $\Omega_t(x,y)$ represents the total score of error for discriminating *(x,y)* as shadow at time instant *t*. Finally, a thresholding operation is applied on $\Omega_t(x,y)$ to determine whether the pixel *(x,y)* belongs to foreground object or cast shadow region.

5 The Shadow Removal Algorithm Based on Dynamic Background

Unlike the first algorithm that removes shadow from foreground mask region, the second algorithm tries to directly classify shadow as background using a dynamic background model. The modeling of background image is a critical issue for resisting camera noise and illumination change. The second algorithm models the dynamic natures of each pixel by maintaining a sorted list of nodes with features and weights. Fig. 2 depicts a flowchart of the proposed shadow removal algorithm based on dynamic background.

The background model of a pixel consists of a sorted list of nodes. The *i*-th node contains two fields: a feature vector m_i and a weighting value w_i. The feature vector $m_i=(r,g,d_h,d_v)$ is composed of the within-pixel invariants *(r,g)* and the between-pixel invariants (d_h,d_v) as defined in section 3. The bigger the weight w_i, the higher the probability of m_i being a feature vector belonging to the background. Suppose the length of the list is *p*, all nodes in a list are sorted in decreasing order according to their weights.

To classify a pixel as foreground/background, the *Mahalanobis distance* between the pixel feature *M* and the feature *m* in each node in the list is calculated according to the following equation:

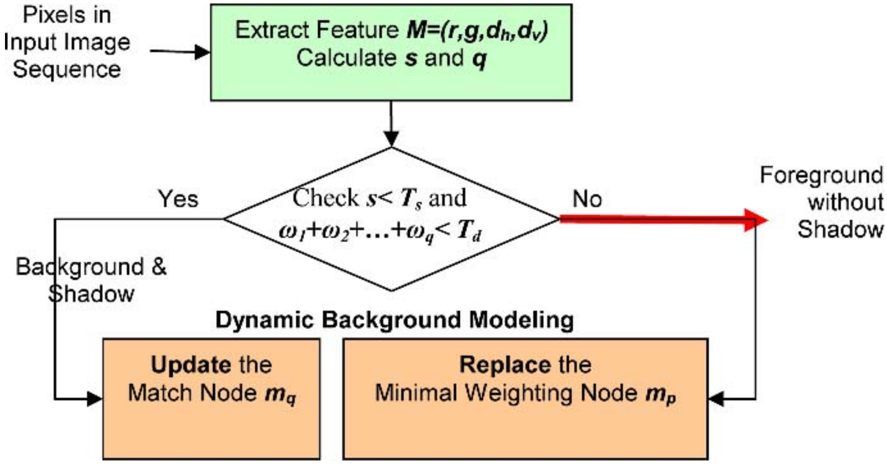

Pixels in Input Image Sequence
Extract Feature M=(r,g,d_h,d_v)
Calculate s and q
Check s< T_s and
ω_1+ω_2+…+ω_q< T_d
Yes
No
Foreground without Shadow
Background & Shadow
Dynamic Background Modeling
Update the Match Node m_q
Replace the Minimal Weighting Node m_p

Fig. 2. Flowchart of the second shadow removal algorithm for dynamic background

$$
\begin{gathered}
\|M-m_i\| = \min_{1\le i\le p}\left[(M-m_i)^T\Sigma^{-1}(M-m_i)\right] \\
\begin{cases} s=\min\limits_{1\le i\le p}\|M-m_i\| \\ q=\arg\min\limits_{1\le i\le p}\|M-m_i\| \end{cases}
\end{gathered}
\tag{12}
$$

where Σ is the covariance matrix that is calculated in advance. Suppose the q-th node is the best match with the minimal distance s. If the minimal distance s is lower than a threshold T_s and the sum of weights of the first q nodes $\omega_1+\omega_2+\ldots+\omega_q$ is lower than a dynamic threshold T_d, then the pixel is classified as background. Otherwise, the pixel is marked as foreground. To exploit spatial consistency, the dynamic threshold T_d is equal to a constant $T_c \in [0,\ldots,1]$ multiplied by an adaptive background probability that is determined by the newest classification results of pixels in a neighborhood window W.

$$
T_d = T_c \cdot \frac{\sum\limits_{(x,y)\in W} F(x,y)}{\sum\limits_{(x,y)\in W} 1}
\tag{13}
$$

$$
F(x,y)=\begin{cases} 1 & \text{if } (x,y) \text{ is classified as background (or shadow) in last frame} \\ 0 & \text{if } (x,y) \text{ is classified as foreground in last frame} \end{cases}
$$

If the pixel is marked as foreground, the node with the lowest weight is replaced with the pixel feature M, i.e., $m_p=M$. If the pixel is classified as background, the best matching node m_q is adaptively updated with the following recursive equation:

$$
m_q = \alpha \cdot M + (1-\alpha)\cdot m_q
\tag{14}
$$

where α is a learning rate which can be decided empirically according to the motion in the scene. Similarly, the weighting value w can be updated by the following equation:

$$w_i = \begin{cases} \beta + (1-\beta)\cdot w_i & \text{if } i = q \\ (1-\beta)\cdot w_i & \text{if } 1 \le i \le p \text{ and } i \ne q \end{cases} \tag{15}$$

where β is another learning rate which can be decided empirically according to the scene. The ranges of α and β are between 0 and 1. After the updating, the sum of all the weights in the list should be equal to one and the nodes should be sorted again in decreasing order according to their new weights.

Since the feature vectors in the list consist of texture/color invariants that are independent of shadow, the algorithm should be able to directly classify shadow pixels as background. As a result, the detected foreground does not include cast shadow. The shadow removal and foreground/background segmentation are fused together in this algorithm.

6 Experimental Results

Several indoor and outdoor scenarios have been tested using the proposed algorithms. Two scenarios are discussed due to space limitation. The image sequence of the first scenario is borrowed from an indoor human tracking experiment [11]. In fig. 3(a), a person walks across a room with his shadow casted on a door. Fig. 3(b) shows the results of Toth's method. A white region shows the foreground, a gray region indicates the cast shadow, and a black region represents the background. It can be

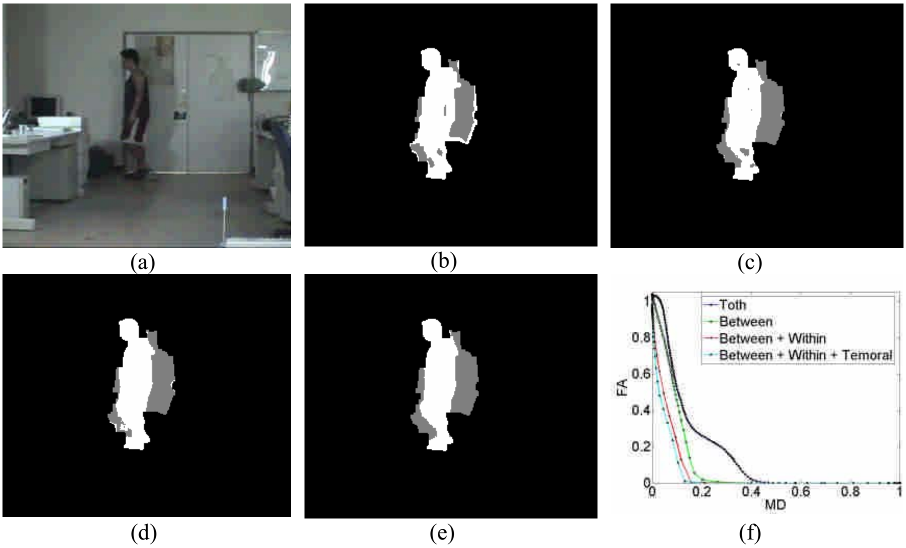
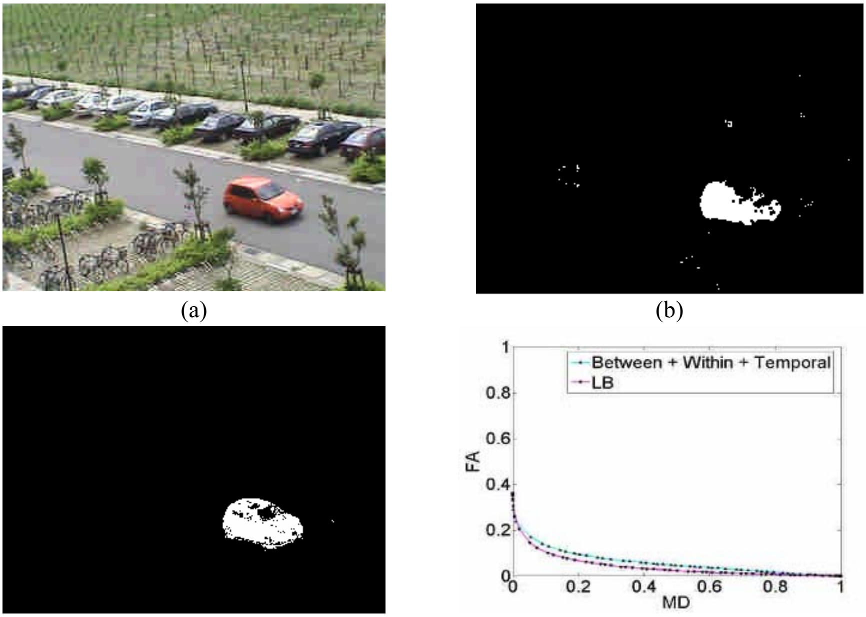


Fig. 3. The experimental results of an indoor human tracking scenario. (a) An input frame. (b) Results of Toth's method. (c) Results with between-pixels invariants. (d) Results with between-pixel & within-pixel invariants. (e) Results with between-pixel, within-pixel invariants & temporal consistency. (f) Comparison of four methods by measuring the ROCs.

observed that the rim of the shadow region (penumbra) is wrongly classified as foreground. Fig. 3(c) demonstrates the results of the first algorithm using only between-pixel invariants. The proposed method tends to correctly classify penumbra as shadow, while Toth's method cannot deal with penumbra properly since their error score is significantly larger especially when the penumbra region becomes broader or the neighborhood window *W* becomes bigger. There are a few holes inside the person indicating that they have been wrongly classified as shadow (false alarm). The reason is that both the clothes and background lack of texture and cannot be discriminated without color information. Fig. 3(d) demonstrates the results exploiting both between-pixel and within-pixel invariants. With the color information combined, detection performance is much better except a small region around the door knob where specular reflection dominates its appearance. This kind of outliers can be removed by utilizing the temporal consistency as shown in Fig. 3(e). Fig. 3(f) plots the performance of the system using ROC curves. The ground truth of shadow regions is marked manually to calculate the false alarm and miss detection rate. The cyan curve with triangle marks (Δ) shows the results of the proposed system considering all factors (within-pixel, between-pixel invariants, and temporal consistency), the red curve with plus marks (+) indicates the results considering both the within-pixel and between-pixel invariants, the green curve with circle marks (o) indicates the results considering only the between-pixels invariants, and the blue curve with star marks (*) represents the results using the shading constraints in Toth's method. It should be noted that only shading information is considered and no morphological filtering is applied in this comparison for all the methods.



(a) (b)

(c) (d)

Fig. 4. The experimental results of an outdoor vehicle tracking scenario. (a) An input frame. (b) Results of the first algorithm. (c) Result of the second algorithm. (d) Comparison of the ROCs.

The second scenario, a car casting shadow on the ground moves across an outdoor scene with waving trees and grasses. Fig. 4(a) shows an input frame in the image sequence. Fig. 4(b) demonstrates the results of the first algorithm in that many false alarms appear due to the dynamic nature of the scenario. As shown in Fig. 4(c), the second algorithm generates better results since the variations of tree and grass pixels are effectively maintained and updated by the dynamic background model. Fig. 4(d) compares the ROC curves of the proposed algorithms. The cyan curve with triangle marks (Δ) shows the results of the first algorithm based on static background and the purple curve with inverse triangle marks (∇) indicates the results of the second algorithm considering dynamic background. The execution rates of both algorithms are around 30 frames per second. Generally speaking, the first algorithm is more efficient and is suitable for indoor scene with static background. The second algorithm is more reliable in outdoor natural scenes with significant perturbations.

7 Conclusions

This paper proposed two reliable and efficient moving cast shadow removal algorithms that combine color/texture invariants and spatial-temporal consistency based on static and dynamic background respectively. The experimental results showed that the proposed algorithms can remove penumbra as well as umbra in several indoor and outdoor scenarios under various illumination conditions.

Acknowledgements

This research was supported in part by the National Science Council of Taiwan.

References

1. Cucchiara, R., Grana, C., Piccardi, M. & Prati, A.: Detecting Moving Objects, Ghosts, and Shadows in Video Streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Pages:1337-1342, Oct. (2003)
2. Fung, G., Yung, N., Pang, G. & Lai, A.: Towards Detection of Moving Cast Shadows for Visual Traffic Surveillance. *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 4, Pages:2505-2510, Oct. (2001)
3. Funt, B. & Finlayson, G.: Color Constant Color Indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Issue 5, Pages:522-529 (1995)
4. Heikkila, M. & Pietikainen, M.: A Texture-based Method for Modeling the Background and Detecting Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 4 (2006)
5. Hsieh, J., Yu, S., Chen, Y. & Hu, W.: A Shadow Elimination Method for Vehicle Analysis. *International Conference on Pattern Recognition* (2004)
6. Siala, K., Chakchouk, M., Besbes, O. & Chaieb, F.: Moving Shadow Detection with Support Vector Domain Description in the Color Ratios Space. *International Conference on Pattern Recognition*, Vol. 4, Pages:384-387 (2004)

7. Stauder, J., Mech, R. & Ostermann, J.: Detection of Moving Cast Shadows for Object Segmentation. *IEEE Transactions on Multimedia*, Pages:65-76 (1999)
8. Toth, D., Stuke, I., Wagner, A. & Aach, T.: Detection of Moving Shadows using Mean Shift Clustering and a Significance Test. *International Conference on Pattern Recognition*, Vol. 4, Pages:260-263, Aug. (2004)
9. Wang, J., Chung, Y., Chang, C. & Chen, S.: Shadow Detection and Removal for Traffic Images. *International Conference on Networking, Sensing and Control*, Vol. 1, Pages:649-654, Mar. (2004)
10. Xu, D., Liu, J., Liu, Z. & Tang, X.: Indoor Shadow Detection for Video Segmentation. *IEEE International Conference on Multimedia and Expo.* (2004)
11. Yang, M., Wang, S. & Lin, Y.: A Multi-modal Fusion System for People Detection and Tracking. *International Journal of Imaging System and Technology*, Vol. 15, Issue 2, Pages:131-142 (2005)

A Unified Approach for Combining ASM into AAM

Jaewon Sung and Daijin Kim

Department of Computer Science and Engineering, POSTECH, Korea
{jwsung, dkim}@postech.ac.kr

Abstract. Since the goal of Active Appearance Model (AAM) is to minimize the residual error between the model appearance and the input image, it often fails to converge accurately to the landmark points of the input image. To alleviate this weakness, we have combined Active Shape Model (ASM) into AAM, where ASM tries to find correct landmark points using the local profile model. Because the original objective function and search scheme of the ASM is not appropriate for combining these methods, we modified the objective function of the ASM and proposed a new objective function that combining that of two methods. The proposed objective function can be optimized using a gradient based algorithm as in the AAM. Experimental results show that the proposed method reduces the average fitting error when compared with existing fitting methods such as ASM, AAM, and Texture Constrained-ASM (TC-ASM).

1 Introduction

Since Active Shape Model (ASM) [1] and Active Appearance Model (AAM) [2], [3] were introduced, many researchers have focused on these methods to solve many image interpretation problems, especially for facial and medical images [4], [5], [6]. Until now, ASM and AAM have been treated as two independent methods in most cases even though they share the same underlying statistical models of the shape and appearance for the target objects (here, the term *appearance* is used with a somewhat broad meaning; it can represent the whole texture or the local texture). However, the two methods cannot be easily combined because they had different optimization goals and used different optimization techniques[7]:

1. ASM models the image texture only in the neighboring region of each landmark point, whereas AAM uses the appearance of the whole image region.
2. ASM finds the best matching points by searching the neighboring region of the current shape positions, whereas AAM compares its current model appearance with the appearance sampled at the current shape positions in the image.
3. ASM seeks to minimize the distance between model points and the identified match points, whereas AAM minimizes the difference between the synthesized model appearance and the target image.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 353–362, 2006.

We found some approaches that tried to improve the performance of each algorithm by adding the features of the other method. The existing intensity-based AAM has some drawbacks: it is sensitive to changes in lighting conditions and it fails to discriminate noisy flat textured area and real structure, and thus may not lead to accurate fitting in AAM search. To alleviate this problem, Scott et al. [8] and Cootes et al [9] proposed augmenting the appearance model of the AAM using some nonlinear factors such as cornerness, edgeness, and gradient directions. Yan et al [10] proposed the TC-ASM that inherited the ASM's local appearance model because of its robustness to varying light conditions. They also borrowed the AAM's global texture, to act as a constraint over the shape and providing an optimization criterion for determining the shape parameters. In TC-ASM, the conditional distribution of a shape parameter given its associated texture parameter was modeled as a Gaussian distribution and there was a linear mapping $\boldsymbol{s}_t = \mathbf{R}\boldsymbol{t}$ between the texture $\boldsymbol{t}$ and its corresponding shape $\boldsymbol{s}_t$, where $\mathbf{R}$ is a projection matrix that can be pre-computed from the training pairs $\{(\boldsymbol{s}_i, \boldsymbol{t}_i)\}$. The search stage computes the next shape parameters by interpolating the shape from a traditional ASM search and the texture-constrained shape. Using the texture constrained shape enabled the search method to escape from the local minima of the ASM search, resulting in improved fitting results.

In this paper, we propose a new fitting method that integrates AAM and ASM in a unified gradient-based optimization framework. One simple and direct combination of ASM and AAM is to alternate between them. In this case, the parameters may not converge to a stable solution because they use different optimization goals and techniques. To guarantee stable and precise convergence, we changed the profile search step of the ASM to a gradient-based search like the AAM search method and combined the error terms of the AAM and ASM into a single objective function. Experimental results show that the proposed fitting method successfully improves the fitting results in terms of root mean squared (RMS) positional errors compared to ASM, AAM, and TC-ASM.

This paper is organized as follows. Section 2 briefly reviews the shape and appearance models in ASM and AAM. Section 3 explains the proposed fitting method that incorporates the shape and appearance models of ASM and AAM in a unified framework. Section 4 presents the experimental results and discussion. Finally, Section 5 presents our conclusion.

2 Background

Assume a set of landmarked face images $D = \{I_i, \boldsymbol{v}_i\}_{i=1}^N$, where N is the number of images, I_i is the i-th image, and $\boldsymbol{v}_i = (x_1, y_1, \ldots, x_v, y_v)^t \in R^{2v \times 1}$ are the coordinates of the landmark points for I_i.

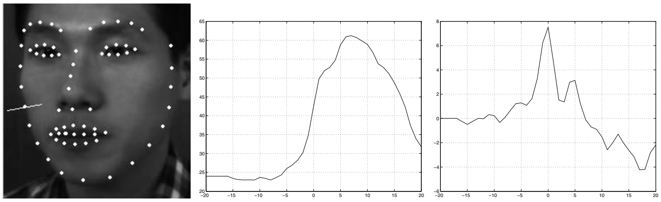
2.1 Shape Model

In ASM and AAM, a shape $\boldsymbol{s} = (x_1, y_1, \ldots, x_v, y_v)^t$ is represented as a linear combination of the mean shape $\boldsymbol{s}_0$ and n orthonormal bases $\boldsymbol{s}_i$ as $\boldsymbol{s} = \sum_{i=0}^n p_i \boldsymbol{s}_i$, where p_i is the ith shape parameter and $p_0 = 1$. These shape bases are obtained by collecting a set of shape vector $\{\boldsymbol{v}_i\}_{i=1}^{N}$, aligning them by removing the variations due to scaling, rotation, and translation, and applying PCA to the resultant aligned shape vectors. The global transformation of the synthesized shape is represented by a vector $\boldsymbol{q} = (q_1, q_2, q_3, q_4)^t$ to describe scaling, rotation, and translation [3].

2.2 Appearance Models

ASM and AAM use different appearance models: the ASM uses a local profile model and the AAM uses a whole appearance model. We introduce these different appearance models in this section.

Local profile model. ASM represents the local appearance at each model point by the intensity or gradient profile ([1]). In this work, we consider the gradient profile because it is not sensitive to global intensity variations. For each landmark point in the training image, the gradient profile $\boldsymbol{G}^{(i,j)}$ $(i = 1, \ldots, N$ and $j = 1, \ldots, v)$, which is a derivative of the intensity profile is obtained. Fig. 1 illustrates an example: (a) landmarks and a normal vector direction at a specific landmark point, (b) the intensity profile, and (c) the gradient profile.



(a) (b) (c)

Fig. 1. An example of intensity profile (b) and gradient profile (c) along the normal direction indicated in (a)

The gradient profile $\boldsymbol{g}^j$ of the j-th model point is also represented as a linear combination of a mean gradient profile $\boldsymbol{g}_0^j$ and l orthonormal gradient profile basis vectors $\boldsymbol{g}_i^j$ as $\boldsymbol{g}^j = \sum_{i=0}^{l} \beta_i^j \boldsymbol{g}_i^j$, where β_i^j is the i-th gradient profile parameter and $\beta_0^j = 1$. The gradient profile basis vectors are obtained by collecting a set of gradient profile vectors $\{\boldsymbol{G}^{(i,j)}\}_{i=1}^{N}$, and applying PCA to them. Fig. 2 shows a mean and the first three gradient profile basis vectors.

Whole appearance model. In AAM ([2], [3]), the whole appearance is defined on the mean shape $\boldsymbol{s_0}$ and the appearance variation is modeled by the linear combination of a mean appearance A_0 and m orthonormal appearance basis vectors A_i as $A(\boldsymbol{x}) = \sum_{i=0}^{m} \alpha_i A_i(\boldsymbol{x})$, where α_i is the i-th appearance parameter and $\alpha_0 = 1$. The appearance basis vectors are computed by applying PCA to the shape normalized appearance images that are warped to a mean shape $\boldsymbol{s}_0$.

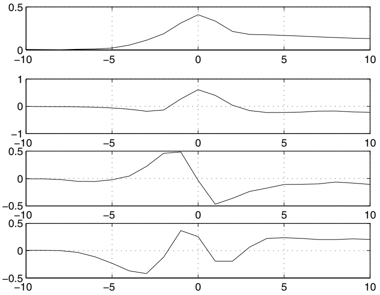


Fig. 2. A mean and the first three gradient profile basis vectors

3 A Unified Approach

3.1 Integrated Objective Function

The objective function of the proposed method consists of three error terms: the error of whole appearance E_{aam}, the error of local appearance E_{asm}, and a regularization error E_{reg}. The last error term is introduced to prevent the shape parameters from deviating too widely. We explain each error term and then introduce the overall objective function that consists of the three error terms.

First, we define the error of the whole appearance model for AAM as

$$E_{aam}(\boldsymbol{\alpha}, \boldsymbol{p}, \boldsymbol{q}) = \frac{1}{N} \sum_{\boldsymbol{x} \in \boldsymbol{s}_0} \left[\sum_{i=0}^{m} \alpha_i A_i(\boldsymbol{x}) - I(W(\boldsymbol{x}; \boldsymbol{p}, \boldsymbol{q})) \right]^2, \tag{1}$$

where N is the number of the pixels $\boldsymbol{x} \in \boldsymbol{s}_0$, and $\boldsymbol{\alpha}$, $\boldsymbol{p}$, and $\boldsymbol{q}$ are the appearance, shape, and similarity transformation parameters. Second, we define the error of local appearance model for ASM as

$$\begin{aligned} E_{asm}(\boldsymbol{\beta}, \boldsymbol{p}, \boldsymbol{q}) &= \frac{K}{v \cdot N_{pf}} \sum_{j=1}^{v} \sum_{z} E_{asm}^{j}(z)^2 \\ &= \frac{K}{v \cdot N_{pf}} \sum_{j=1}^{v} \sum_{z} \left\{ \sum_{i=0}^{l} \beta_i^j \boldsymbol{g}_i^j(z) - I(W^j(z; \boldsymbol{p}, \boldsymbol{q})) \right\}^2, \end{aligned} \tag{2}$$

where N_{pf} is the length of the gradient profile vector, K is a scaling factor to balance the magnitude of E_{asm} with that of the E_{aam}, β_i^j is the i-th gradient profile model parameter corresponding to the j-th model point, and $W^j(z; \boldsymbol{p}, \boldsymbol{q}) = \boldsymbol{s}^j(\boldsymbol{p}, \boldsymbol{q}) + z\boldsymbol{n}^j$ represents a warping function that transforms a scalar coordinate z of the 1-D gradient profile vector into a 2D image coordinate of the image to be used for reading the image gradient profile $\boldsymbol{g}$ at each j-th model point; The $\boldsymbol{s}^j(\boldsymbol{p}, \boldsymbol{q})$ and $\boldsymbol{n}^j$ are the j-th model point of the current shape corresponding to current shape parameters $\boldsymbol{p}$ and $\boldsymbol{q}$, and the normal vector of the j-th model point. Third, we define a regularization error term E_{reg}, which constrains the range of the shape parameters p_i to get a good fitting result, as

$$E_{reg}(\boldsymbol{p}) = R \cdot \sum_{i=1}^{n} \frac{p_i^2}{\sqrt{\lambda_i}^2}, \tag{3}$$

where λ_i is the eigenvalue corresponding to the i-th shape basis $\boldsymbol{s}_i$ and R is a constant that controls the effect of regularization term. If the value of R is set to a large value, the fitting result tends to be close to the mean shape. While the shape parameters are directly limited not to exceed $\pm 3\sqrt{\lambda_i}$ after each iteration in traditional ASM and AAM, we add E_{reg} into the objective function for the same effect.

Combining (1), (2), and (3), we define an integrated objective function E as

$$E = (1-\omega)(E_{aam} + E_{reg}) + \omega E_{asm}, \tag{4}$$

where $\omega \in [0,1]$ determines how significant the E_{asm} term will be in the overall objective function E. So, the proposed algorithm operates like AAM when $\omega = 0$, and like ASM when $\omega = 1$.

3.2 Adaptive Control of E_{asm}

First, we consider the effect of E_{asm} on convergence. The local profile model for the ASM has learned the local profile variations only near the landmark points of the training data. Thus, the E_{asm} term must be controlled for a good model fitting in the following manner. During early iteration, it should have little influence because the synthesized shape is typically far from the landmark points. In later iterations, as the synthesized shape becomes closer to the landmark points, the effect of E_{asm} should become stronger. To reflect this idea, we need a measure to indicate how accurately the model shape is converged to the landmark points. Fortunately, the E_{aam} term meets well this requirement. The degree of convergence is then represented by a bell-shaped function ([11]) of the E_{aam} term:

$$Bell(a,b,c;E_{aam}) = \left(1 + \left|\frac{\sqrt{E_{aam}} - c}{a}\right|^{2b}\right)^{-1}, \tag{5}$$

where a, b, and c parameters determine the width of the bell, the steepness of downhill curve, and the center of the bell, respectively. In this work, we set $c = 0$ to use the right side of the bell-shape.

Second, we consider how well each model point has converged to its landmark point. Although the synthesized shape is converged to the landmark points on average, some points are close to their landmark points but other points are still far from their landmark points. To accommodate this situation, we consider a weighting function $exp\left(-\frac{E_{asm}^j}{2\sigma_j}\right)$, where $E_{asm}^j (j = 1, \ldots, v)$ is the local profile error at j-th model point, and σ_j controls the sensitivity of this weighting function.

Considering these two effects for controlling the E_{asm} term, the original weight ω is modified as

$$\omega^j = \omega \cdot Bell(E_{aam}) \cdot exp\left(-\frac{E^j_{asm}}{2\sigma_j}\right), \tag{6}$$

where $\omega^j (j = 1, \ldots, v)$ denotes the effective weight of the jth model point.

4 Experiment Results and Discussions

4.1 Database

We used our own face database that consists of 80 face images, which were collected from 20 people with each person having 4 different expressions (neutral, happy, surprised and angry). All 80 images were manually landmarked. The shape, appearance, and gradient profile basis vectors were constructed from the images using the methods explained in Section 2. Fig. 3 shows some typical images in the face database.

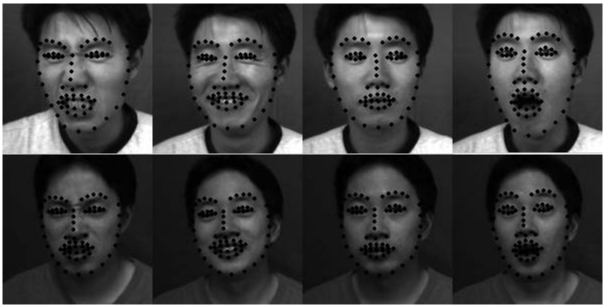


Fig. 3. A set of example face images

4.2 Fitting Performance

First, we determined the optimal number of the linear gradient profile basis vectors. For this, we built a linear gradient model using 40 images of 10 randomly selected people, fitted the generated model to them, and measured the average fitting error of 70 landmark points, where the fitting error was defined by the distance between a landmark point and its converged vertex.

Fig. 4 shows the average fitting error, where the $*$ denotes the mean value of average fitting error when AAM was used, the $\circ$ denotes the mean value of the average fitting error against the number of linear gradient profile basis vectors when gradient-based ASM was used, and the bar denotes the standard deviation of the average fitting error in both methods. This figure shows that (1) the average fitting error of gradient-based ASM is smaller than that of AAM, (2) the optimal number of the linear gradient profile basis vectors is 7, and (3) the corresponding average fitting error is approximately 0.5 pixel. Thus, the number of linear gradient profile basis vectors was 7 in our experiments.

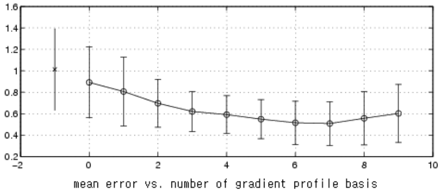


Fig. 4. Mean and standard deviation of average fitting errors

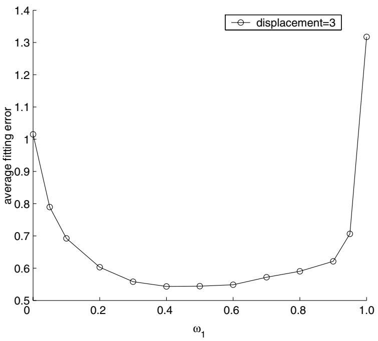


Fig. 5. The effect of ω on the average fitting error

Second, we investigated the effect of the E_{asm} term on the fitting performance. By setting the value of ω to 0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, and 1.0. The scale factor K was set to 10,000 to make the magnitude of the E_{asm} term similar to that of the E_{aam} term (the order of E_{aam} was 10 and the order of E_{asm} was 10^{-3}). For the 40 training images used in the first experiment, the optimal similarity transform parameters were computed from the landmark points, and then the position of the initial shape was moved 3 pixels in a random direction. The initial shape parameters are set to zero.

Fig. 5 shows the average fitting error at each ω, where the case of $\omega = 0$ corresponds to the AAM and the case of $\omega = 1.0$ corresponds to the gradient-based ASM. This figure shows that (1) the average fitting error of the AAM could be minimized further by choosing an optimal value of the ω that incorporates the effect of gradient-based ASM and (2) the smallest average fitting error was achieved when the ω was set to about 0.5.

Fig. 6 illustrates a typical example of fitted results when the AAM and the AAM+ASM (ω=0.5) were used. In this figure, the white and black dots correspond to the fitted results of AAM and AAM+ASM, respectively. The AAM+ASM converged more accurately to the landmark points than the AAM, particularly near the mouth and chin.

Third, we compared the fitting performance of AAM+ASM with existing methods such as the traditional AAM and TC-ASM, another approach combining AAM and ASM. We set $\omega = 0$ for AAM and $\omega = 0.5$ for AAM+ASM. We built three linear models (appearance, shape, and profile) using the 40 training

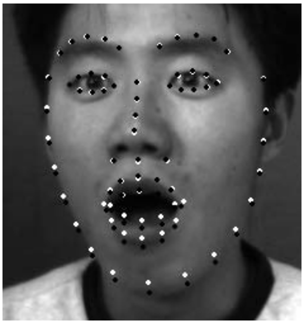


Fig. 6. A typical example of fitted results

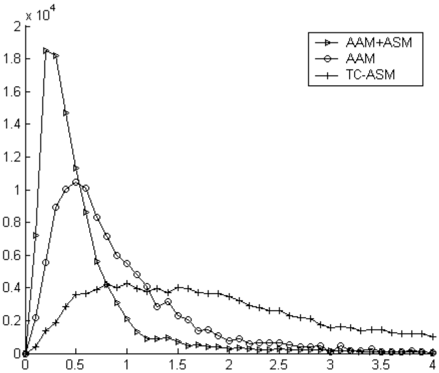


Fig. 7. A histogram of fitting errors

images that were used in previous experiments and we measured the fitting performance using the remaining 40 test images. For one test image, we tried 40 different initial positions, where they are corresponding to 5 different distances (3, 5, 7, 9, and 11) and 8 different directions. The initialization was done as follows: the shape and appearance parameters were set to zero, and the scale and position parameters were computed from the landmark points.

Fig. 7 shows a histogram of fitting error for 112,000 cases: 40 images × 40 initial positions × 70 vertices. This figure shows that the AAM+ASM produced the smallest mean and standard deviation of fitting error.

Fig. 8 shows the convergence rates of the different methods in terms of different initial displacements for three different threshold values. Here, we assume that the fitting is converged when the average fitting error is less than the given threshold value. This figure shows that (1) the convergence rate increases as the threshold value increases for all methods, (2) the convergence rate decreases as the initial displacement increases in the AAM+ASM and the AAM methods, (3) the fitting error is almost constant as initial displacement increases in the

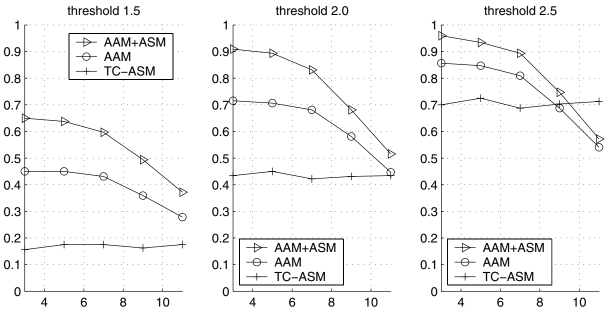


Fig. 8. Convergence rate of the three methods

TC-ASM method because this method employs a search-based ASM, and (4) the convergence rate of the AAM+ASM is the highest among three methods in almost all cases.

5 Conclusion

In this paper, we have proposed a unified gradient based framework that combines ASM into AAM and also proposed an adaptive weight control strategy that improved the stability of convergence. Originally, AAM used the whole appearance model and a gradient based approach for model fitting, while ASM used a local profile model and a search based approach for model fitting. Since these properties were not appropriate for combination, we introduced the gradient based approach for ASM. Basically, AAM+ASM method worked similarly to AAM method and it had an additive property that guaranteed more precise convergence to the landmark points by reducing the fitting error due to the incorporated profile error term.

Currently, we have to manually determine a set of parameters such as ω and σ^j to obtain the best performance, and their optimal values may be different from one set of data and another. In the future, we will try to develop more generally applicable methods by designing parameter-free weight control mechanisms. Furthermore, it may be possible to implement the proposed algorithm more efficiently by incorporating the gradient-based ASM search in the inverse compositional AAM fitting method.

Acknowledgement

This work was partially supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei University. Also, it was partially supported by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

References

1. Cootes, T., Cooper, D., Taylor, C., and Graham, J. 1995. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38-59.
2. Cootes, T., Edwards, G., and Taylor, C. 2001. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681-685.
3. Matthews, I. and Baker, S. 2004. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135-164.
4. Lanitis, A., Taylor, C., and Cootes, T. 1997. Automatic interpretation and aoding of face images using flexible models. *IEEE Trans. on Pattern Analysis and Maching Intelligence*, 19(7):743-756.
5. Kuilenburg, H., Wiering, M., and Uyl, M. 2005. A model based method for automaic facial expression recognition. in *European Conference on Maching Learning.*
6. Ginneken, B., Frangi, A., Staal, J., Romeny, B., and Viergeber, M. 2002. Active shape model segmentation with optimal features. *IEEE Trans. on Medical Imaging*, 21(8):924-933.
7. Cootes, T., Edwards, G., and Taylor, C. 1999. Comparing active shape models with active appearance models. in *British Machine Vision onference.*
8. cott, I., Cootes, T., and Taylor, C. 2003. Improving appearance odel matching using local image structure, in *Conference on nformation Processing in Medical Imaging*, 2732:258-269.
9. Cootes T. and Taylor, C. 2001. On representing edge structure for model matching. in *Conference on Computer Vision and Pattern Recognition*, 1:1114-1119.
10. Yan, S., Liu, C., Li, S., Zhang, H., Shum, H., and Cheng, Q. 2002. Texture-constrained active shape models. in *European Conference on Computer Vision.*
11. Jang, J., Sun, C., and Mizutani, E. 1997. Neuro-Fuzzy and Soft Computing, *Prentice Hall.*

Variational Approach to Cardiac Motion Estimation for Small Animals in Tagged Magnetic Resonance Imaging

Hsun-Hsien Chang[1,2]

[1] Department of Electrical and Computer Engineering
[2] Pittsburgh NMR Center for Biomedical Research
Carnegie Mellon University, Pittsburgh, PA 15213, USA
hsunhsien@cmu.edu

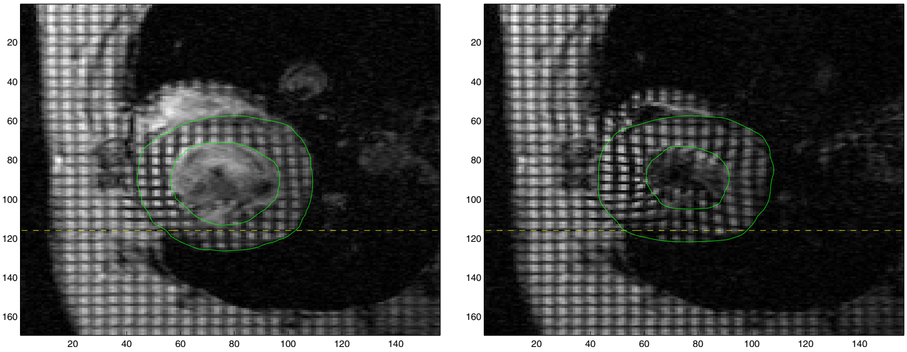
Abstract. Monitoring cardiac motion in the stage of small animal study is very important in cardiac research. This paper presents a variational approach to estimating the heart motion of small animals imaged by magnetic resonance (MR) tagging. Small animals have much faster heart beats than human, so their cardiac sequences are temporally undersampled, leading to the aperture problem when reconstructing the cardiac motions. To overcome this difficulty, we adopt the prior knowledge of motions on the myocardial boundaries that were determined in the preprocessing. In addition, we utilize the high gradients of intensities on the tag lines to derive the motions through the cardiac cycle. We formulate the problem in the framework of energy minimization. Variational calculus gives us the Euler-Lagrange equations to seek the minimum. The results produced by our approach are better than the existing optical flow based method [1] and the harmonic phase method [2]. The evaluation suggests that our approach will be more suitable for the small animal studies.

1 Introduction

Small animal study is an important stage in cardiac research. In this research stage, cardiologists use rats, rodents, mice, or rabbits to investigate the cardiac effects of new drugs; i.e., they monitor myocardial motion with respect to drugs. The advent of tagged magnetic resonance imaging (MRI) [3] enables cardiologists to monitor the dense motion of the heart. Tagged MRI generates cardiac images with tag lines superimposed onto the myocardium. These tag lines are anchored to the heart tissue so they deform consistently with the cardiac motion, see Figure 1. Tracking tag lines leads to tracking the heart motion.

Many algorithms have been proposed to process the tagged data of the human heart. Gupta and Prince [1] adopt the framework of optical flow to estimate tag motions. Chen and Amini [4] use B-spline to detect the tag lines and then infer tag motions. Osman and Prince [2] develop the harmonic phase (HARP) method which tracks the phase of tag lines through the cardiac cycle in the frequency domain. Unfortunately, these algorithms are not applicable to small

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 363–372, 2006.



(a) Frame 1.

(b) Frame 2.

Fig. 1. Tagged cardiac MR images of a rat at frames 1 and 2. The myocardium has been segmented.

animal images. The only algorithm developed specifically for small animals is by Chang *et al.* [5], who adopt fiber biomechanics and intensity constancy in an optimization scheme. However, this algorithm is not robust.

Small animals have much faster heart beats than human, so the MRI data is temporally undersampled. With reference to Figure 1(a) where the heart is at end-diastole, we draw a dotted line corresponding to a horizontal tag line. When the heart contracts to the next phase shown in Figure 1(b), the fixed dotted line in the myocardium is closer to another tag line than to the original one. Thus, the existing algorithms will recognize the wrong tag line, and hence produce the incorrect motions. Such a phenomenon is the aperture problem in the intensity-based methods and is the phase-wrapping problem in the frequency-based approaches. To overcome the difficulties raised by temporal undersampling, we need different strategies to handle small animal images.

Given a sequence of tagged MR images, we first manually segment the endocardium and epicardium through the cardiac cycle. We adopt the framework of energy minimization to estimate dense cardiac motions. Our objective energy functional consists of four terms that are described in the following.

1. The first term is formulated from data information which considers the features of tag lines. The intensities of tag lines fade away over time due to the relaxation of MR signals through the cardiac cycle [3], but the contrast between tagged and untagged pixels is still relatively high. Unlike Gupta and Prince's approach [1] that introduces a set of parameters describing the intensity fading, we utilize the high gradients to track the heart.
2. The second term is derived from the motions on the segmented endocardium and epicardium and on the background. We preprocess the segmented contours to obtain the motions at the heart boundaries. The boundary motions will induce other myocardial pixels to move consistently. On the other hand, the background in the image is static and should have no motion. We treat the boundary motions and the static background as prior knowledge in the objective functional.

3. The third term enforces the motion field to be smooth as possible, because the heart is a continuum. This term penalizes the gradients of the heart motions.
4. The last term prevents the jerky segmentation that will affect our prior knowledge of motions in the second term. In other words, we require the contours to be smooth by minimizing their length.

The organization of this paper is as follows. Section 2 derives the objective energy functional. Section 3 describes our minimization scheme. Section 4 presents the experimental results. Finally, we conclude this paper in Section 5.

2 Formulation of the Objective Functional

Before going through the details of our objective functional, we introduce the notation. Let Ω and T be the 2D spatial and temporal domains, respectively, of the images. The image intensity is a function $f(x, y, t)$ on $\Omega \times T$. Our goal is to estimate for each pixel $\mathbf{x} = (x, y)$ at time t a motion vector $\mathbf{u}(x, y, t) = [u(x, y, t), v(x, y, t)]^T$, where u and v are the x and y, respectively, components of $\mathbf{u}$.

Our objective functional $J(\mathbf{u})$ to be minimized with respect to the unknown $\mathbf{u}(x, y, t)$ is defined as:

$$J(\mathbf{u}) = \lambda_1 J_1(\mathbf{u}) + \lambda_2 J_2(\mathbf{u}) + \lambda_3 J_3(\mathbf{u}) + \lambda_4 J_4(\mathbf{u}) \; . \tag{1}$$

The first term $J_1(\mathbf{u})$ tries to match the gradients of pixels through the cardiac cycle. The second term $J_2(\mathbf{u})$ treats the boundary motions and static background as constraints. The third term $J_3(\mathbf{u})$ requires the smooth motion field. The last term $J_4(\mathbf{u})$ considers the smoothness of segmented endocardial and epicardial contours. Finally, the λ_i's weight appropriately the different terms. We now detail each energy term in the following subsections.

2.1 $J_1(\mathbf{u})$: Gradient Constancy

The tag lines produce high intensity contrast to the neighboring pixels. This fact can be captured by matching the gradients of the images through the cardiac cycle. Considering a small time step δt, we obtain the equation of constant gradients as

$$\nabla f(x, y, t) = \nabla f(x + u\delta t, y + v\delta t, t + \delta t) \; . \tag{2}$$

Using Taylor expansion, we obtain the gradient conservation equations:

$$(\nabla f_x)^T \mathbf{u} + \frac{\partial f_x}{\partial t} = 0 \tag{3}$$

$$(\nabla f_y)^T \mathbf{u} + \frac{\partial f_y}{\partial t} = 0 \; , \tag{4}$$

where $f_x = \frac{\partial f}{\partial x}$ and $f_y = \frac{\partial f}{\partial y}$. In fact, the conservation equations (3) and (4) are affected by noise; that is, the observation from the data has errors ε_1 and ε_2

$$\varepsilon_1 = (\nabla f_x)^T \mathbf{u} + \frac{\partial f_x}{\partial t} \tag{5}$$

$$\varepsilon_2 = (\nabla f_y)^T \mathbf{u} + \frac{\partial f_y}{\partial t} \; . \tag{6}$$

In the sequel, our goal is to minimize the quadratic errors, and the first energy term $J_1(\mathbf{u})$ is defined as

$$J_1(\mathbf{u}) = \int_{\Omega} \left[(\nabla f_x)^T \mathbf{u} + \frac{\partial f_x}{\partial t} \right]^2 + \left[(\nabla f_y)^T \mathbf{u} + \frac{\partial f_y}{\partial t} \right]^2 d\Omega \; . \tag{7}$$

2.2 $J_2(\mathbf{u})$: Prior Knowledge of Motions

In the preprocessing, we segment the endocardial and epicardial boundaries, denoted by C_{en} and C_{ep} respectively, and then determine their motions. We further introduce a level set function $\phi(x, y)$ on the image, so the contours C_{en} and C_{ep} are represented by pixels with zero level sets. We determine the level set function $\phi(x, y)$ as

$$\phi(x, y) = \begin{cases} + \min(d(\mathbf{x}, C_{\mathrm{en}}), d(\mathbf{x}, C_{\mathrm{ep}})), & \text{if } \mathbf{x} \text{ is in the heart} \\ - \min(d(\mathbf{x}, C_{\mathrm{en}}), d(\mathbf{x}, C_{\mathrm{ep}})), & \text{if } \mathbf{x} \text{ is not in the heart} \end{cases} , \tag{8}$$

where $d(\mathbf{x}, C)$ means the closest distance between the given pixel $\mathbf{x}$ and the contour C.

To mask out the myocardium, the myocardial boundaries, and the background, we further define the regularized Heaviside function

$$H(\phi) = \frac{1}{2} \left[1 + \frac{2}{\pi} \arctan \left(\frac{\phi}{\epsilon} \right) \right] \; . \tag{9}$$

The regularized Heaviside function represents the pixels in the myocardium, and $1 - H(\phi)$ is the function representing the pixels in the background. The function $|\nabla H(\phi)|$ approximately represents the pixels at the myocardial boundaries C_{en} and C_{ep} [6].

Assume that there is a motion field $\mathbf{u}^{\mathrm{pr}}$ whose values on the myocardial boundaries C_{en} and C_{ep} are predetermined. To utilize the prior knowledge of boundary motions, we consider the quadratic errors between the unknown motion map $\mathbf{u}$ and the prior $\mathbf{u}^{\mathrm{pr}}$ on the zero level sets:

$$\int_{\Omega} ||\mathbf{u} - \mathbf{u}^{\mathrm{pr}}||^2 |\nabla H(\phi)| d\Omega \; . \tag{10}$$

On the other hand, the background in the cardiac MR images is static. This fact is equivalent to minimizing the magnitude of the motions outside the heart,

$$\int_{\Omega} ||\mathbf{u}||^2 \left[1 - H(\phi) \right] d\Omega \; . \tag{11}$$

Considering equations (10) and (11) together leads to our second energy term $J_2(\mathbf{u})$:

$$J_2(\mathbf{u}) = \int_\Omega ||\mathbf{u} - \mathbf{u}^{\mathrm{pr}}||^2 |\nabla H(\phi)| + ||\mathbf{u}||^2 \left[1 - H(\phi)\right] d\Omega \tag{12}$$

$$= \int_\Omega \left[(u - u^{\mathrm{pr}})^2 + (v - v^{\mathrm{pr}})^2\right] \delta(\phi)|\nabla\phi| + (u^2 + v^2)\left[1 - H(\phi)\right] d\Omega, \tag{13}$$

where

$$\delta(\phi) = \frac{dH(\phi)}{d\phi} = \frac{1}{\pi}\left(\frac{\epsilon}{\epsilon^2 + \phi^2}\right) . \tag{14}$$

is the regularized Dirac function.

2.3 $J_3(\mathbf{u})$: Smoothness of Motions

The heart is a continuum, so its motion should be smooth. The smoothness constraint penalizes the gradient of the heart motions and gives us the third energy term:

$$J_3(\mathbf{u}) = \int_\Omega ||\nabla\mathbf{u}||^2 H(\phi) d\Omega = \int_\Omega (u_x^2 + u_y^2 + v_x^2 + v_y^2) H(\phi) d\Omega , \tag{15}$$

where $u_x = \frac{\partial u}{\partial x}$, $u_y = \frac{\partial u}{\partial y}$, $v_x = \frac{\partial v}{\partial x}$, and $v_y = \frac{\partial v}{\partial y}$.

2.4 $J_4(\mathbf{u})$: Smoothness of the Myocardial Boundaries

The segmented boundaries of the heart should be smooth. This is equivalent to minimizing the length of the contours. Hence, the fourth energy term is

$$J_4(\mathbf{u}) = \int_\Omega |\nabla H(\phi)| d\Omega = \int_\Omega \delta(\phi)|\nabla\phi| d\Omega . \tag{16}$$

3 Solutions to Motion Estimation

In the fourth energy term J_4, smoothing the myocardial boundaries will deform the level set function ϕ. This implies that our objective is a functional not only of $\mathbf{u}$ but also of ϕ. Summing all the energy terms leads to our objective functional

$$J(\mathbf{u}, \phi) = \lambda_1 J_1(\mathbf{u}, \phi) + \lambda_2 J_2(\mathbf{u}, \phi) + \lambda_3 J_3(\mathbf{u}, \phi) + \lambda_4 J_4(\mathbf{u}, \phi) \tag{17}$$

$$\begin{aligned} &= \lambda_1 \int_\Omega \left[(\nabla f_x)^T \mathbf{u} + \frac{\partial f_x}{\partial t}\right]^2 + \left[(\nabla f_y)^T \mathbf{u} + \frac{\partial f_y}{\partial t}\right]^2 d\Omega \\ &\quad + \lambda_2 \int_\Omega \left[(u - u^{\mathrm{pr}})^2 + (v - v^{\mathrm{pr}})^2\right] \delta(\phi)|\nabla\phi| + (u^2 + v^2)\left[1 - H(\phi)\right] d\Omega \\ &\quad + \lambda_3 \int_\Omega (u_x^2 + u_y^2 + v_x^2 + v_y^2) H(\phi) d\Omega \\ &\quad + \lambda_4 \int_\Omega \delta(\phi)|\nabla\phi| d\Omega . \end{aligned} \tag{18}$$

Applying calculus of variations [7], the functions u, v, ϕ that minimize $J(\mathbf{u}, \phi)$ in equation (17) must satisfy the Euler-Lagrange equations:

$$\begin{aligned} 0 = {} & \lambda_1(f_{xx}u + f_{xy}v + f_{xt})f_{xx} + \lambda_1(f_{yx}u + f_{yy}v + f_{yt})f_{yx} \\ & + \lambda_2(u - u^{\mathrm{pr}})\delta(\phi)|\nabla\phi| + \lambda_2 u\,[1 - H(\phi)] - \lambda_3\nabla^2 u H(\phi)\,, \end{aligned} \tag{19}$$

$$\begin{aligned} 0 = {} & \lambda_1(f_{xx}u + f_{xy}v + f_{xt})f_{xy} + \lambda_1(f_{yx}u + f_{yy}v + f_{yt})f_{yy} \\ & + \lambda_2(v - v^{\mathrm{pr}})\delta(\phi)|\nabla\phi| + \lambda_2 v\,[1 - H(\phi)] - \lambda_3\nabla^2 v H(\phi)\,, \end{aligned} \tag{20}$$

$$\begin{aligned} 0 = {} & \big[-\lambda_2(u^2 + v^2) + \lambda_3(u_x^2 + u_y^2 + v_x^2 + v_y^2) \\ & - \big(\lambda_2(u - u^{\mathrm{pr}})^2 + \lambda_2(v - v^{\mathrm{pr}})^2 + \lambda_4\big)\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\bigg]\,\delta(\phi)\,. \end{aligned} \tag{21}$$

We recursively solve for u, v, ϕ in the Euler-Lagrange equations (19), (20), and (21) by superscripting them a pseudo-time index k. The recursive equations are

$$\begin{aligned} u^{(k)} - u^{(k+1)} = {} & \lambda_1^{(k)}(f_{xx}u^{(k)} + f_{xy}v^{(k)} + f_{xt})f_{xx} + \lambda_1^{(k)}(f_{yx}u^{(k)} + f_{yy}v^{(k)} + f_{yt})f_{yx} \\ & + \lambda_2^{(k)}(u^{(k)} - u^{\mathrm{pr}})\delta(\phi^{(k)})|\nabla\phi^{(k)}| + \lambda_2^{(k)}u^{(k)}\left[1 - H(\phi^{(k)})\right] \\ & - \lambda_3^{(k)}\nabla^2 u^{(k)} H(\phi^{(k)})\,, \end{aligned} \tag{22}$$

$$\begin{aligned} v^{(k)} - v^{(k+1)} = {} & \lambda_1^{(k)}(f_{xx}u^{(k)} + f_{xy}v^{(k)} + f_{xt})f_{xy} + \lambda_1^{(k)}(f_{yx}u^{(k)} + f_{yy}v^{(k)} + f_{yt})f_{yy} \\ & + \lambda_2^{(k)}(v^{(k)} - v^{\mathrm{pr}})\delta(\phi^{(k)})|\nabla\phi^{(k)}| + \lambda_2^{(k)}v^{(k)}\left[1 - H(\phi^{(k)})\right] \\ & - \lambda_3^{(k)}\nabla^2 v^{(k)} H(\phi^{(k)})\,, \end{aligned} \tag{23}$$

$$\begin{aligned} \phi^{(k)} - \phi^{(k+1)} = {} & \delta(\phi^{(k)})\Big[-\lambda_2^{(k)}(u^{(k)^2} + v^{(k)^2}) + \lambda_3^{(k)}(u_x^{(k)^2} + u_y^{(k)^2} + v_x^{(k)^2} + v_y^{(k)^2}) \\ & - \Big(\lambda_2^{(k)}(u^{(k)} - u^{\mathrm{pr}})^2 + \lambda_2^{(k)}(v^{(k)} - v^{\mathrm{pr}})^2 + \lambda_4^{(k)}\Big)\operatorname{div}\left(\frac{\nabla\phi^{(k)}}{|\nabla\phi^{(k)}|}\right)\Bigg]\,. \end{aligned} \tag{24}$$

When attempting to estimate the heart motion by minimizing energy functional (17), we encounter an important issue on *how to choose the appropriate values for the weighting parameters* λ_1 *to* λ_4. Their relative values emphasize differently each of the energy terms. The larger value a weighting parameter has, the more its energy term dominates during minimization. To determine the weightings λ_i, we adopt the technique of annealing schedule [8]. The annealing is to change the values of λ_i's over k. Our strategy emphasizes the data term J_1 and the prior knowledge term J_2 in the beginning, and then in the end focuses on the smoothness terms J_3 and J_4. We achieve this strategy by setting the weighting parameters as follows:

$$\lambda_1^{(k)} = 1 - \frac{0.5}{\cosh(5\pi(\frac{k}{N} - 1)} \ , \tag{25}$$

$$\lambda_2^{(k)} = 0.5 + 0.5\cos\left(\frac{k\pi}{2N}\right) \ , \tag{26}$$

$$\lambda_3^{(k)} = 0.2 + 0.8\frac{k}{N} \ , \tag{27}$$

$$\lambda_4^{(k)} = \frac{1}{\cosh(5\pi(\frac{k}{N} - 1))} \ , \tag{28}$$

where N is the number of iterations. Figure 2 visualizes the values of λ_i's versus the iterations.

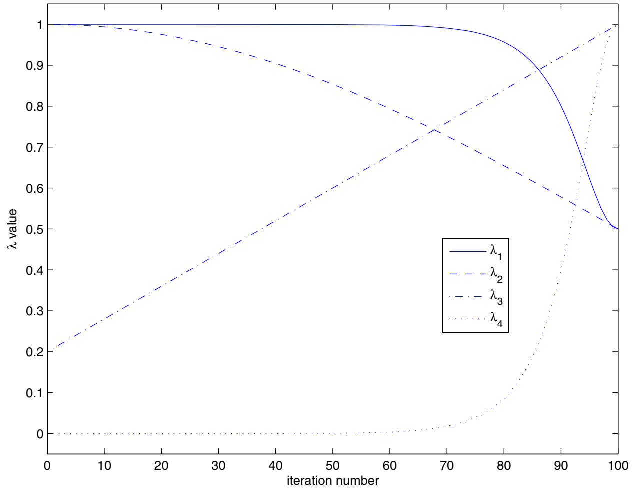
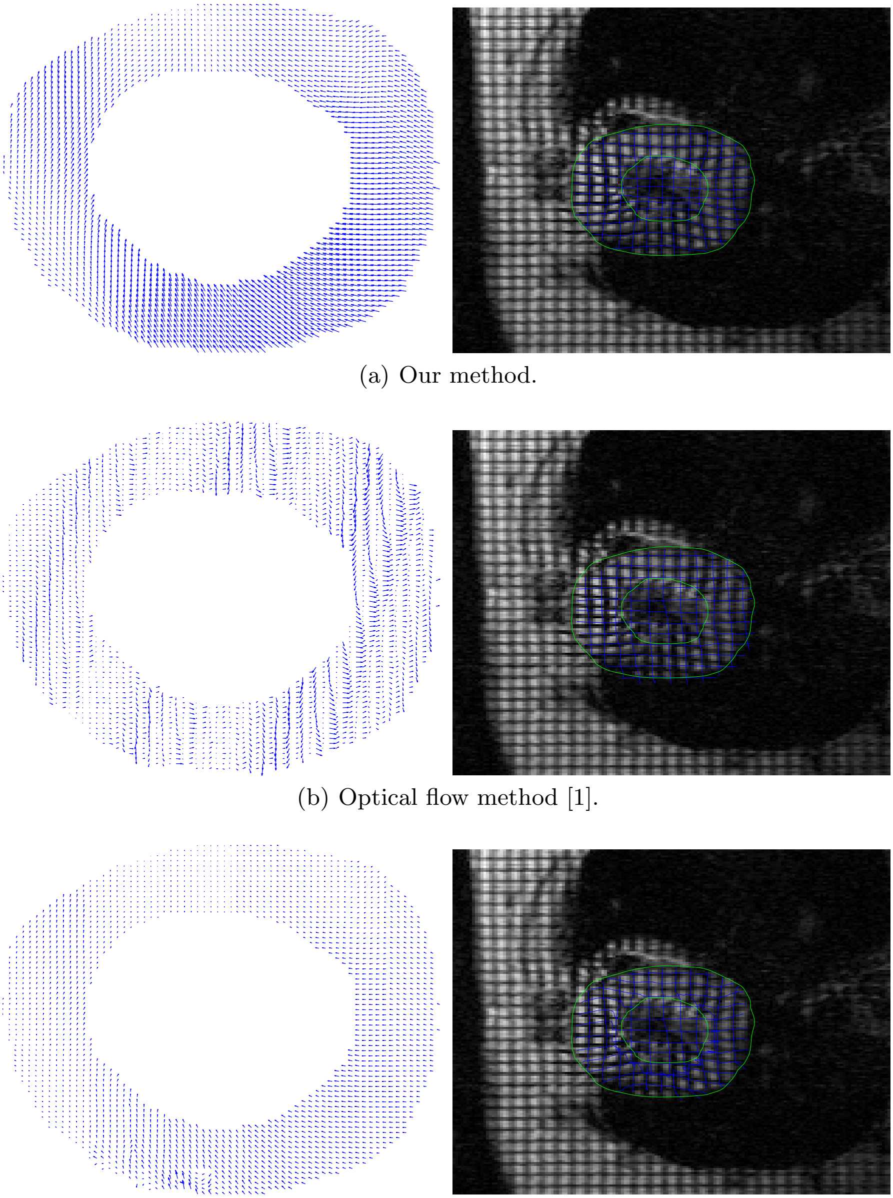


Fig. 2. Weighting parameters λ_i

4 Experiments

To demonstrate the efficacy of our approach, we apply our algorithm to cardiac MRI data of rats. ECG and respiration gated cine MRI generate images with resolution of $156\mu m \times 156\mu m$. MR tagging was achieved by a modified DANTE sequence. We cover the heart at 10 time phases through the cardiac cycle. All MRI scans were performed on a Bruker AVANCE DRX 4.7-T system. All the algorithms are implemented with MATLAB®.

The estimated motion map obtained by our approach for Figures 1(a) and 1(b) is shown in Figure 3(a). In this figure, we also display where the tag lines appear at frame 2 by applying estimated motions to them. On the other hand, we also apply the optical flow based algorithm [1] and HARP method [2] to the same



(a) Our method.

(b) Optical flow method [1].

(c) HARP method [2].

Fig. 3. Left: Motion map estimated between frames 1 and 2. Right: The deformation of the tag lines estimated at frame 2.

cardiac sequence; their results are shown in Figures 3(b) and 3(c), respectively. From the experimental results, we can clearly see that our method can superiorly overcome the difficulties that arise from the temporal undersampling, while other methods fail.

To evaluate the performance of our algorithm, we use manual tracking as the ground truth. We label carefully the locations of tag lines through the entire sequence of the cardiac images. From the manually detected tag lines, we can derive the motions of myocardial pixels. To compare quantitatively different approaches, we compute the mean square deviations of the motion maps from the ground truth. The smaller the deviations, the better the approach is. Figure 4 shows the errors at all the frames. From the comparisons, our method always has the least errors, while optical flow based approach has the largest.

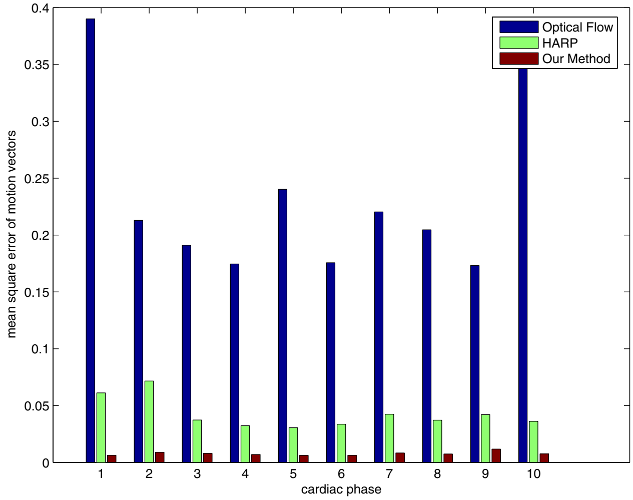


Fig. 4. Mean square errors of motion maps using optical flow approach, HARP algorithm, and our method

5 Conclusions

This paper develops a variational approach to estimating cardiac motion for small animals imaged by tagged MRI. In small animal studies, we face the temporally undersampled images, which cause the aperture problem. To overcome this difficulty, we segment the heart in the preprocessing and treat the motions of the segmented boundaries as prior knowledge. We also consider the high gradients on the tag lines, the smoothness of the motion field, and the smoothness of the segmented boundaries. Using the energy minimization framework, we derived the motion estimates. The experimental results demonstrate that our algorithm is superior to other methods and is useful for the small animal studies.

Acknowledgments

The author would like to thank Dr. Yijen L. Wu for providing the data sets and Dr. Sheng I. Lin for discussing the research ideas.

References

1. Gupta, S.N., Prince, J.L.: On variable brightness optical flow for tagged MRI. In: Proceedings of Information Processing in Medical Imaging, Ile de Berder, France (1995) 323–334
2. Osman, N.F., McVeigh, E.R., Prince, J.L.: Imaging heart motion using harmonic phase MRI. IEEE Transactions on Medical Imaging **19**(3) (2000) 186–202
3. Zerhouni, E.A., Parish, D.M., Rogers, W.J., Yang, A., Shapiro, E.P.: Human heart: tagging with MR imaging–a method for noninvasive assessment of myocardial motion. Radiology **169** (1988) 59–63
4. Chen, Y., Amini, A.A.: A MAP framework for tag line detection in SPAMM data using Markov random fields on the B-spline solid. IEEE Transactions on Medical Imaging **21**(9) (2002) 1110–1122
5. Chang, H.H., Moura, J.M.F., Wu, Y.L., Sato, K., Ho, C.: Reconstruction of 3-D dense cardiac motion from tagged MR sequences. In: Proceedings of IEEE International Symposia on Biomedical Imaging, Arlington, VA (2004) 880–883
6. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. International Journal of Computer Vision **50**(3) (2002) 271–293
7. Smith, D.R.: Variational Methods in Optimization. Dover, Mineola, NY (1974)
8. Pluempitiwiriyawej, C., Moura, J.M.F., Wu, Y.L., Ho, C.: STACS: New active contour scheme for cardiac MR image segmentation. IEEE Transactions on Medical Imaging **24**(5) (2005) 593–603

Performance Assessment of Image Fusion*

Qiang Wang and Yi Shen

Harbin Institute of Technology, Harbin, 150001, P.R.China
wangqiang@hit.edu.cn

Abstract. Performances evaluation method that can compare and analyze different fusion techniques is an essential part of image fusion techniques. In this paper we propose a performance assessment method for image fusion techniques based on accurate measurement of general relationship among a image set. Numerical verifications, using data constructed from four kinds of typical relations and multi-variable time series generated from a logistic function, are conducted to demonstrate the proposed concept of nonlinear correlation information entropy and its characteristics. Furthermore, the performances of two widely used image fusion techniques, i.e. wavelet transform based fusion and pyramid transform based fusion operating on typical hyperspectral image sets, are evaluated using the proposed method. The performances evaluation results agree with the classification accuracy in application.

1 Introduction

As the image fusion techniques have been developing quickly in a number of applications such as remote sensing [1], medical imaging [2], digital camera vision [3], and military applications [4] in recent years, the methods that can assess or evaluate the performances of different fusion technologies objectively, systematically, and quantitatively have been recognized as an urgent requirement. Xydeas and Petrovic [5] have proposed a framework for measuring objectively pixel-level image fusion performance, which is regarded as a perceptual means. Qu and Zhang [6] and Ramesh and Ranjith [7] have proposed a measure for evaluating the image fusion performance by using mutual information. Wang and Shen [8] have proposed a Quantitative Correlation Analysis (QCA) method to evaluate the performances of hyperspectral image fusion techniques. A fast method for QCA have also been proposed, which can fulfill the same task with a faster speed comparing to the original QCA method, especially when the number of source images increases and the size of the image expands [9].

Both the perceptual means in [5] and the mutual information based means in [6∼7] can only assess the image fusion techniques that just fuse two source image into one image. The QCA and fast QCA method can assess the image fusion techniques that have multi-input source images and multi-output images, but they assess the performances just according to the linear correlation between

* Project Supported by Development Program for Outstanding Young Teachers in Harbin Institute of Technology.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 373–382, 2006.

the source images and the fused images[8∼9]. Therefore, in some applications, the evaluation results are not as accurate as enough.

In a typical image fusion process, assume that we have P source images $S = \{S_i; i = 1, 2, \ldots, P\}$ and Q fusion techniques $T = \{T_i; i = 1, 2, \ldots, Q\}$. Each fusion technique T_i will fuse source images S into M_i result images $F_i = \left\{F_i^j; j = 1, 2, \ldots, M_i\right\}$. All Fused images are $F = \{T_i(S); i = 1, 2, \ldots, Q\} = \{F_i^j; i = 1, 2, \ldots, Q; j = 1, 2, \ldots, M_i\}$

The general relationship among source images S is fixed after we chose them. Let original images and different fused images F_i construct different systems $Y_i = \{S_1, S_2, \ldots, S_N, F_i\}$. Therefore, the relationships among the systems Y_i are different because of differences of the fused images F_i. If relationship of the system Y_i is stronger than that of the system Y_j, it indicates the fused image F_i has more correlation with the source images than F_j has. It also indicates the fusion technique T_i has the ability of fusing more information from original images to the fused image. So the fusion technique T_i is better than T_j from the viewpoint of information usage.

To measure the nonlinear relationship among the multi variables of the system Y_i, the concept of nonlinear correlation coefficient and nonlinear correlation information entropy are defined in Section 2. Section 3 presents numerical verifications using constructed time series of known origin to prove the definition and statistics of the proposed concept. Experiments on using the proposed evaluation measure to assess two types of widely used image fusion techniques (Wavelet Transform Based Fusion and Pyramid Transform Based Fusion), which in the paper fuse a typical hyperspectral image set, are conducted in Section 4. Section 5 discusses the relation with and differences from other measures proposed in the research field. Finally, conclusions are given in Section 6.

2 Nonlinear Correlation Information Entropy

For describing the general correlation between two variables while not only linear correlation as the correlation coefficient does, the mutual information concept is widely used. Mutual information can be thought of as a generalized correlation analogous to the linear correlation coefficient, but sensitive to any relationship, not just linear dependence [10]. However, it can be seen from the definition of the mutual information that it does not ranges in a definite closed interval as the correlation coefficient does, which ranges in [0, 1] with 0 indicates the minimum linear correlation and 1 indicates the maximum.

Considering two discrete variables $X = \{x_i\}_{1 \le i \le N}$ and $Y = \{y_i\}_{1 \le i \le N}$, they are firstly resorted in ascending order and placed into b ranks with first N/b samples in the first rank, the second N/b samples in the second rank, and so on. Secondly, the sample pairs$\{(x_i, y_i)\}_{1 \le i \le N}$, are placed into a $b \times b$ rank grids by comparing the sample pairs to the rank sequences of X and Y.

The revised joint entropy of the two variables X and Y is defined as

$$H^r(X,Y) = -\sum_{i=1}^{b}\sum_{j=1}^{b} \frac{n_{ij}}{N} \log_b \frac{n_{ij}}{N} \tag{1}$$

where n_{ij} is the number of samples distributed in the ij^{th} rank grid. Nonlinear Correlation Coefficient (NCC) is defined as

$$NCC(X;Y) = H^r(X) + H^r(Y) - H^r(X,Y) \tag{2}$$

where $H^r(X)$ is the revised entropy of the variable X, which is defined as

$$H^r(X) = -\sum_{i=1}^{b} \frac{n_i}{N} \log_b \frac{n_i}{N}. \tag{3}$$

Notice that the number of samples distributed into each rank of X and Y is invariant, and the total number of sample pairs is N, so equation (2), can be rewritten as

$$NCC(X;Y) = 2 + \sum_{i=1}^{b^2} \frac{n_i}{N} \log_b \frac{n_i}{N}. \tag{4}$$

NCC not only is sensitive to the nonlinear correlation of two variables, but also can describe this relationship with a number ranges from the closed interval [0,1], with 0 indicates the minimum general correlation and 1 indicates the maximum one. In the maximum correlation condition, sample sequences of the two variables are exactly the same, i.e. $x_i = y_i (i = 1, 2, \ldots, N)$. So $NCC(X;Y) = 2 + \sum_{i=1}^{b^2} p_i \log_b p_i = 2 + b \times \frac{N/b}{N} \log_b \frac{N/b}{N} = 1$. Under the minimum correlation situation, while the sample pairs distributed equally into the $b \times b$ ranks. So $NCC(X;Y) = 2 + \sum_{i=1}^{b^2} p_i \log_b p_i = 2 + b^2 \times \frac{N/b^2}{N} \log_b \frac{N/b^2}{N} = 0$

For multivariate situation, the general relation between every two variables can be obtained according to the definition of NCC, thus the nonlinear correlation matrix of the K concerned variables can be written as

$$R^N = \{NCC_{ij}\}_{1 \le i \le K, 1 \le j \le K} \tag{5}$$

where NCC_{ij} denotes NCC of the i^{th} and j^{th} variable. As a variable is completely the same as itself, $NCC_{ij} = 1, (i = j, 1 \le i \le K, 1 \le j \le K)$. The diagonal element of R^N, $r_{i,j} = 1, (i = j, i \le K, j \le K)$, represents the autocorrelation of each variable. The rest elements of R^N, $0 \le r_{i,j} \le 1, (i \ne j, i \le K, j \le K)$, denotes the correlation of the i^{th} and j^{th} variable. When the variables have no relation with each other, R^N is unit matrix. In this case, the multi variables have the weakest relation. On the contrary, when all the variables have the strongest correlation with each other, each element of R^N equals to 1. In this situation, the correlation of the multi variables is also the strongest. The general relation of the concerned K variables is implied in R^N. In order to quantitatively measure it, the nonlinear joint entropy H_{R^N} is defined as following

$$H_{R^N} = -\sum_{i=1}^{K} \frac{\lambda_i^{R^N}}{K} \log_K \frac{\lambda_i^{R^N}}{K} \tag{6}$$

where $\lambda_i^{R^N}$ $(i = 1, 2, \ldots, K)$ are the eigenvalues of the nonlinear correlation matrix. According to matrix eigenvalues theory, it can be educed that $0 \leq \lambda_i^{R^N} \leq K(i = 1, 2, \ldots, K)$ and $\sum_{i=1}^{K} \lambda_i^{R^N} = K$. Nonlinear Correlation Information Entropy (NCIE) I_{R^N}, used as a nonlinear correlation measure of the concerned variables, is defined as

$$I_{R^N} = 1 - H_{R^N} = 1 + \sum_{i=1}^{K} \frac{\lambda_i^{R^N}}{K} \log_K \frac{\lambda_i^{R^N}}{K} \tag{7}$$

NCIE has the following characters (mathematical properties): 1), it remains unchanged when the positions of the K variables are changed; 2), it ranges from a closed interval [0,1], with 0 indicates the minimum nonlinear correlation among the variables concerned, while 1 indicates the maximum; 3), it is sensitive to the general relations of the variables concerned, not merely the linear relations.

3 Numerical Verification of NCIE

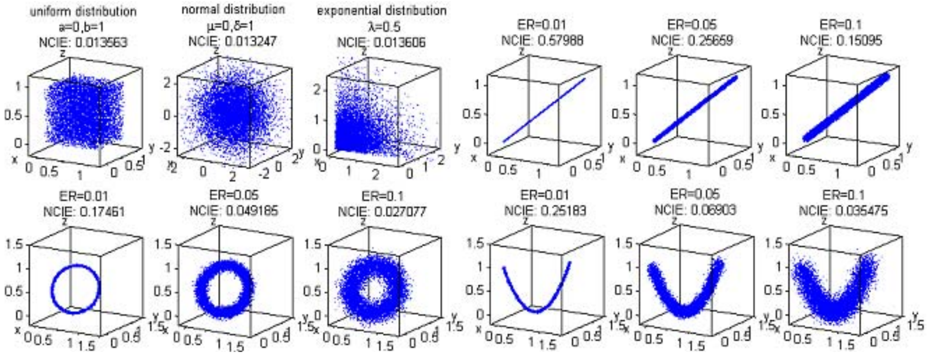
Data constructed from four kinds of typical relations, i.e., random, linear, circular and square functions, and multi-variable time series generated from a logistic function, are used as examples of numerically generated data of the known origin in order to demonstrate the proposed concept of NCIE and its characteristics.

a) The first simulation is conducted on four classical relations displayed in Fig. 1, in which we test the relations of three variables for visualization's concern, although NCIE can be applied to the relation of any number variables. From Fig. 1 we can find that, for variables of three random distributions, their relationship is generally weak, so their NCIE are also very little. For relations of three common functions, i.e. linear, circle and square, noises with different amplitudes are added to generate data of different correlation degrees. As the amplitude of the added noise increases, the correlation degree of the concerned three variables decreases, and their NCIE also decreases. This result conforms to our definition of NCIE, which states that larger NCIE indicates stronger correlation. Meanwhile, the tendency that stronger correlation has larger NCIE can be obviously found.

b) A logistic equation, which is thoroughly studied and applied in [10], is used to generate the time series

$$x_t = 4x_{t-4}(1 - x_{t-4}) \tag{8}$$

The reason for the use of this logistic equation is that the distinct peaks in the lagged self-mutual information can be obtained. The logistic equation produces



uniform distribution a=0,b=1 NCIE: 0.013563

normal distribution μ=0,δ=1 NCIE: 0.013247

exponential distribution λ=0.5 NCIE: 0.013606

ER=0.01 NCIE: 0.57988

ER=0.05 NCIE: 0.25659

ER=0.1 NCIE: 0.15095

ER=0.01 NCIE: 0.17461

ER=0.05 NCIE: 0.049185

ER=0.1 NCIE: 0.027077

ER=0.01 NCIE: 0.25183

ER=0.05 NCIE: 0.06903

ER=0.1 NCIE: 0.035475

Fig. 1. NCIE of three random variables of three classical distributions, i.e. uniform distribution, normal distribution and exponential distribution, and three common relations, i.e. linear relation, circular relation and square relation with different noises added to the variables. The number of samples of each variable $N = 10000$ and the bin number $b = 100$.

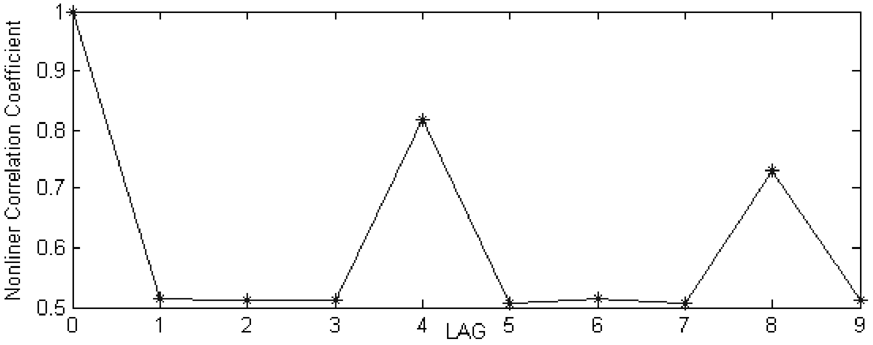


Fig. 2. Nonlinear correlation coefficient between the original time series and time series lagged by $i (0 \leq i \leq 9)$ steps from the original

a chaotic time series and each series in the simulation has 1000 values. NCC between the generated time series and a version of itself lagged by k time steps is computed for k ranging from 0 to 9 (in simulation the bin number is set to 50), Fig. 2 shows these results.

The simulation of NCIE is conducted on the generated time series. The generated series has maximum correlation with the version of itself lagged by 4 steps. NCIE of some typical combinations of time series are shown in Table 1.

According to Table 1, and noting that a series has maximum correlation with the version of itself lagged by 4 steps, we can find that the general relationships of group 1 to 6 increase gradually, and for group 6, four series are the same, i.e., they are maximally related. The NCIE of group 1 to 6 also increase, and

Table 1. NCIE of some typical combinations of time series

Group ID	combinations	NCIE	Group ID	combinations	NCIE
1	1,2,3,4	0.23732	7	1,2,3,4,5	0.29850
2	1,2,3,5	0.31285	8	1,2,3,4,6	0.29708
3	1,1,2,3	0.41622	9	1,2,3,4,7	0.29774
4	1,2,2,5	0.49498	10	1,2,3,4,8	0.29659
5	1,2,5,5	0.53013	11	1,2,3,4,9	0.27880
6	5,5,5,5	1.0000	12	1,2,3,5,9	0.36321

for group 6, gets to the maximum. Moreover, the correlations in group 7, 8, 9, 10 and 11 are almost equal, and NCIE of these groups are almost equal too. Group 12 contains three most correlated band, i.e. band 1, 5 and 9. Therefore, its correlation is stronger and its NCIE is larger. Comparing group 1 and group 7-11, each of the latter contains one more band, which is maximally correlated with one of the band in its own group. Thus the latter has more correlation than the former. Comparing group 2 with group 7-11, we can find the latter has less correlation than the former and its NCIE is also less. According to the analysis above, it can be concluded that, if the bands in a group have more correlation, NCIE of this group is larger than others. This conclusion completely complies with our definition of NCIE.

4 Experiments

In order to test the proposed method, an experiment is conducted, in which two widely used multi-resolution analysis based image fusion, Wavelet Transform based Fusion (WTF) and Pyramid Transform based Fusion (PTF), are evaluated. The detail information about the fusion techniques can be found in [11]. Our experiments are conducted on AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) data, which is downloaded from LARS (Laboratory for Applications of Remote Sensing) at Purdue University. The data set consists of a portion of an AVIRIS data taken in June 1992, which covers a mixture of agricultural/forestry land in the Indian Pine Test Site in Indiana.

For computational simplicity, ten bands are selected from the 224 bands as source images, which are shown in Fig. 3. The fused images via two fusion methods mentioned above, i.e. WTF and PTF, are displayed in Fig. 4. The performances evaluation and comparison of fusion methods is conducted on the original information (ten source images) and fused information (two fused images). The amount of general relation between source images and fused images respectively is calculated, considering each image as a time series after transforming image matrix into a vector, and results are presented in Table 2.

According to Table 2, we can find that NCIE of WTF result image and ten hyperspectral source images is a little larger than that of PTF result image and source images. That indicates the general relationship between the WTF generated image and source images are stronger than that between the PTF generated

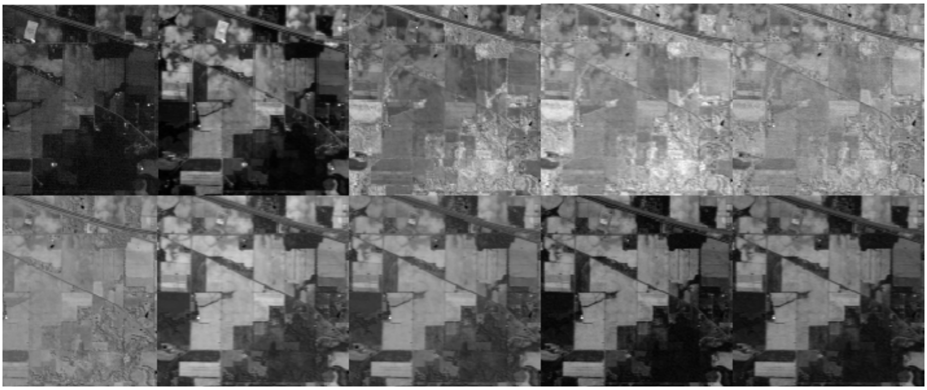


Fig. 3. Original ten images in the experiment

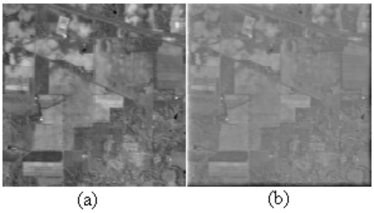


Fig. 4. Fused images by two multi resolution analysis based fusion methods: (a) WTF, (b) PTF

Table 2. Image fusion performances evaluation in the first experiment

Fusion Method	NCIE
Wavelet Transform Based Fusion	0.08700
Pyramid Transform Based Fusion	0.08611

image and source images. So the comparative result of the fusion methods performances can be obtained as following: the better one of the fusion methods based on the efficiency of information usage is the wavelet transform based image fusion method. Its ability to fuse the most information into the result image determines its better performance. The experiment results conform to that of the application conducted in [18], which focuses on the classification of hyperspectral data. Classification accuracy in the application is presented in Table 3.

From Table 3, it can be found that the more accurate classification result is that of wavelet transform based fusion, which performs better in our performance evaluation experiment. It also should be noted that the classification accuracy of WTF is better than that of PTF by just 2.2 percent, which indicates their

Table 3. Hyperspectral image classification Accuracy (%)

	Corn	Grass	Soybean	Forest	Average
Wavelet Transform Based Fusion	97.40	99.60	90.60	99.00	96.65
Pyramid Transform Based Fusion	95.60	97.40	86.40	98.40	94.45

very close performances. The NCIE measures are also very close, but still can reflect their performances differences.

5 Discussions

For assessing performances of different image fusion techniques, Qu[10] and Ramesh[11] have proposed their evaluation measures based on Mutual Information (MI), which is a primary concept in information theory to measure the statistic dependency between two random variables. The measure in [10] is defined as

$$M_F^{AB} = MI_{FA}(f,a) + MI_{FB}(f,b) \tag{9}$$

where

$$MI_{FA}(f,a) = \sum_{f,a} p_{FA}(f,a) \log \frac{p_{FA}(F,a)}{p_F(f)p_A(a)} \tag{10}$$

It refers to the information of fused image F about input image A. $p_F(f)$,$p_A(a)$ are marginal probability density functions, $p_{FA}(f,a)$ is joint probability density function.

The measure in [10] reflects the information that the fused image obtained from both input images A and B. [11] also uses this measure, but call it as Fusion Factor (FF), and states that, larger FF indicates more information has been transferred from two source images to the fused image. Moreover, it points out that, even larger FF still can not indicate the source images are fused symmetrically. Therefore, it develops a concept called Fusion Symmetry (FS) to denote the symmetry of the fusion process about two input images. The less FS is, the better the fusion process performs.

$$FS = |\frac{MI_{AF}}{MI_{AF} + MI_{BF}} - 0.5| \tag{11}$$

From their definitions of above measures we can see that, they are all developed to assess the fusion method that can fuse two source images into one result image. Although multiple source images can be fused recursively, but different fusion sequences will yield different fusion results. Therefore, the measures can not assessing the performances that will fuse any source images in a fusion process. However, for the performances measure proposed in this paper, we can tell from its definition that, it can be used to assess the fusion techniques that may have multi-input source images and multi-output fused images.

To compare the measures, we calculate MI between each fused image and source image of the experiment in section IV. Inspired by [10,11], the sum of MI is calculated as a multi-input version of FF to indicate the information that fused image contains about all input images. The results are presented in Table 4. MI_i refers to MI between the i^{th} input image and the fused image.

Table 4. MI between each source image and fused image of experiment 1

	MI_1	MI_2	MI_3	MI_4	MI_5	MI_6	MI_7	MI_8	MI_9	MI_{10}	FF
WTF	2.5072	3.1854	2.8892	2.8437	2.387	2.4871	3.0939	2.9476	2.3553	2.2770	26.9736
PTF	2.1359	2.7591	2.4743	2.4335	2.0005	2.1303	2.6792	2.5372	1.9837	1.9248	23.0585

As larger FF indicates more information has been fused from source images to the result image, i.e. the fusion techniques performs better, so we can draw the conclusion from Table 4 that WTF outperforms PTF, which conforms to the conclusion in Section IV.

6 Conclusions

As a conclusion, we can say that the proposed nonlinear correlation measure based method can be used to evaluate the performances of different fusion methods. Because of its nature of measuring the general relation among variables, it can give more accurate results. The experiments of performances comparison of WTF and PTF verify the correctness and effectiveness of the proposed measure.

References

1. Terry A. Wilson, Steven K. Rogers.: Perceptual-Based Image Fusion for Hyperspectral Data. IEEE Trans. on Geoscience and Remote Sensing, **35** (1997) 1007–1017
2. Constantinos S. P., Marios S. P.: Medical Imaging fusion applications-An overview. Conference Record of the Asilomar Conference on Signals, Systems and Computers **2** (2001) 1263–1267
3. Zhang Zhong, Blum Rick S.: A Categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application. Proc. of the IEEE **87** (1999) 1315–1326
4. Z. Xue, R. S. Blum.: Fusion of visual and IR images for concealed weapon detection. Proceedings of the Fifth International Conference on Information Fusion (2002) 1198–1205
5. C. S. Xydeas, V. Petrovic.: Objective image fusion performance measure. Electronic Letters **36** (2000) 308–309
6. Guihong Qu, Dali Zhang.: Information measure for performance of image fusion. Electronic Letters **38** (2002) 313–315

7. Chaveli Ramesh, T. Ranjith.: Fusion performance measures and a lifting wavelet transform based algorithm for image fusion. Proceedings of the Fifth International Conference on Information Fusion (2002) 317–320
8. Qiang Wang, Yi Shen.: A quantitative method for evaluating the performances of hyperspectral image fusion. IEEE Trans. on Instrumentation and Measurement **52** (2003) 1041–1047
9. Qiang Wang, Yi Shen.: Fast Quantitative Correlation Analysis and Information Deviation Analysis for Evaluating the Performances of Image Fusion Techniques. IEEE Trans. on Instrumentation and Measurement **53** (2004) 1441–1447
10. Mark S. Roulston.: Significance testing of information theoretic functionals. Physica D **110** (1997) 62–66
11. Terry A. Wilson, Steven K. Rogers.: Perceptual-Based Image Fusion for Hyperspectral Data. IEEE Trans. on Geoscience and Remote Sensing **35** (1997) 1007–1017

Possibilistic C-Template Clustering and Its Application in Object Detection in Images

Tsaipei Wang

Department of Computer Science, National Chiao Tung University,
1001 Ta-Hsieh Rd., Hsinchu 300, Taiwan
wangts@cs.nctu.edu.tw

Abstract. We present in this paper a new type of alternating-optimization based possibilistic c-shell clustering algorithm called possibilistic c-template (PCT). A template is represented by a set of line segments. A cluster prototype consists of a copy of the template after translation, scaling, and rotation transforms. This extends the capability of shell clustering beyond a few standard geometrical shapes that have been studied so far. We use a number of 2-dimensional data sets to illustrate the application of our algorithm in detecting generic template-based shapes in images. Techniques taken to relax the requirements of known number of clusters and good initialization are also described. Results for both synthetic and actual image data are presented.

Keywords: Shell clustering, Fuzzy clustering, Possibilistic clustering, Robust clustering, Object and shape Detection, Template-based methods.

1 Introduction

The ability to efficiently detect shell-like structures of particular shapes is useful in many image and signal processing applications. Fuzzy and possibilistic shell clustering algorithms have been shown to be useful for this purpose. Variations of these algorithms have been developed for different types (shapes) of shell prototypes. Examples include algorithms for the detection of lines and planes [1], circles [2,3], quadratic surfaces [4-6], rectangles [7], and templates [8]. There are also a few reported applications for the detection of lines [9], circles [10], and ellipses [11] in real-world images. Compared with fuzzy clustering algorithms, the possibilistic approach has been shown to be more robust against noise [4,12]. According to [5], fuzzy and possibilistic shell clustering has a number of advantages as compared to generalized Hough transform in detecting particular shapes: It is more computationally efficient without the large memory requirement of Hough transform, is less sensitive to noise and zigzagged edges, and the parameter resolution is not limited by the predefined bin sizes. These properties make shell clustering a useful option for shape/object detection.

Most existing shell clustering algorithms are specifically designed for particular shapes, e.g., circles, hence seriously limiting applications of these algorithms. The only one exception is in [8], which attempted to do shell clustering with

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 383 – 392, 2006.

general-shaped template-based shell prototypes. However, unlike the other algorithms that employ the more efficient and commonly used alternating optimization (AO) approach, [8] relies on genetic algorithm (GA) for optimizing the prototypes. However, GA can take a long time to converge and can still get trapped in local minima of the objective function. The reason of using GA is probably because it does not require update equations of the prototype parameters, as the derivation of these equations is intrinsically difficult for generic templates than for prototypes of simple geometric shapes. (A comparison between fuzzy c-means using only AO or GA can be found in [13].)

This paper is the first presentation of an alternating-optimization scheme applicable to possibilistic shell clustering for template-based prototypes that uses. This extends the applicability of efficient AO-based shell clustering into various tasks where one is looking for surfaces that are not necessarily of basic geometric shapes. The templates are represented as collections of line segments, and prototype parameters consist of parameters of their transforms. By relaxing the common approach of updating all prototype parameters simultaneously, we are able to obtain closed-form update equations, making the process very efficient. In contrast, many existing AO-based shell clustering algorithms have to reply on numerical methods (e.g., [2]) or non-Euclidean distance measures (e.g., [4]) in order to solve their prototype update equations.

2 Possibilistic C-Template Clustering

2.1 Possibilistic C-Means and C-Shell Clustering

The following is the standard objective function used for various types of possibilistic c-means and c-shell clustering algorithms [12]:

$$J=\sum_{j=1}^{C}\sum_{i=1}^{N}u_{ij}^{m}\,d_{ij}^{2}+\sum_{j=1}^{C}\eta_{j}\sum_{i=1}^{N}(1-u_{ij})^{m}\ . \tag{1}$$

Here N is the number of data points, C is the number of clusters, m is the fuzzification factor, u_{ij} is the membership of the i^{th} data point in the j^{th} cluster, and d_{ij} is a distance measure between the i^{th} data point and the j^{th} cluster prototype, and η_j is termed the "bandwidth" in [12] and controls the dependence of u_{ij} on d_{ij}.

The standard alternating-optimization scheme involves iteratively solving the equations that correspond to conditions of local minima of (1): solving $\partial J/\partial u_{ij}=0$ for the update equations for all u_{ij}, and solving $\partial J/\partial \boldsymbol{\theta}_j=0$ for the update equations of all $\boldsymbol{\theta}_j$. Here $\boldsymbol{\theta}_j$ represents the set of parameters used to define the j^{th} cluster prototype. The solution for u_{ij} is given by [12]

$$u_{ij}=\left[1+\left(\frac{d_{ij}^{2}}{\eta_{j}}\right)^{\frac{1}{m-1}}\right]^{-1}\ . \tag{2}$$

Equation (2) is applicable to all possibilistic clustering algorithms. It is the solutions for $\boldsymbol{\theta}_j$ that are difficult. While simple closed-form solution exists for point prototypes when using Euclidean distance, this usually is not the case for shell prototypes. As a result, iterative numerical methods such as Newton's method, as used in [2], become necessary, adding significant amount of computational cost. In [3,4], a non-Euclidean algebraic distance measure is used to obtain closed-form solutions for updating quadratic shell prototypes. While this is computationally efficient, it is applicable only to quadratic-shell clusters, and the non-Euclidean distance measure sometimes yields clustering results that are not intuitively optimal [6]. In contrast, we employ Euclidean distance throughout this paper.

2.2 Shell Clustering with Template-Based Prototypes

Before we can explain our approach of deriving the closed-form update equations for the prototype parameters, we have to define to templates themselves first. In this paper, a template consists of a set of line segments. It can be represented as a set of vertices and edges that connect them:

$$T = \{V, E\} \ , \tag{3}$$

where V is the set of vertices,

$$V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{N_V}\}\,, \tag{4}$$

and E is the set of edges,

$$E = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{N_E}\}\,. \tag{5}$$

Each edge is actually represented by the indices of its starting and ending vertices. N_V and N_E are the number of vertices and edges, respectively. Fig. 1 displays a few templates of different shapes that are used in the experiments in section 3.

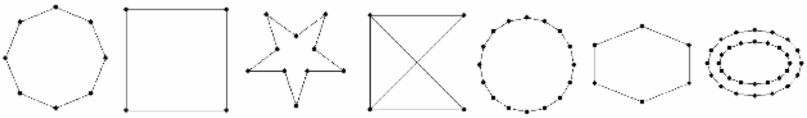


Fig. 1. Templates used in our experiments

The prototype of a cluster is a transformed version of the template. Therefore, the prototype parameter set $\boldsymbol{\theta}_j$ actually consists of parameters that define the transform. While many different types of transforms can be considered here, to limit the complexity of the problem, we start by only considering three shape-preserving transforms: scalar scaling, rotation, and translation. A point on the j^{th} prototype $\boldsymbol{p}$ is related to its corresponding point in the template, $\boldsymbol{p}^*$, according to

$$\boldsymbol{p} = R_j s_j \boldsymbol{p}^* + \boldsymbol{t}_j\,. \tag{6}$$

Here s_j, R_j, and $\boldsymbol{t}_j$ are the scalar scaling factor, rotation matrix, and translation vector of the j^{th} prototype, respectively. For 2-dimensional data, this means that there are 4 adjustable parameters for each cluster prototype. R_j is the 2×2 rotation matrix determined by the rotation angle θ_j. In this paper we only focus on 2-dimensional data. In addition, we do not need to be concerned about transforms of the edges because they are simply determined from the respective vertices.

For each data point $\boldsymbol{x}_i$ ($1 \le i \le N$), we define $\boldsymbol{p}_{ij}$ as the point on the j^{th} prototype that is closest to $\boldsymbol{x}_i$. Therefore, the Euclidean distance between a data point and a cluster is given by

$$d_{ij}{}^2 = \left\| \boldsymbol{x}_i - \boldsymbol{p}_{ij} \right\|^2 = \left\| \boldsymbol{x}_i - \left(R_j s_j \boldsymbol{p}_{ij}^* + \boldsymbol{t}_j \right) \right\|^2 . \tag{7}$$

To obtain the update equations for the prototype parameters, i.e., s_j, R_j, and $\boldsymbol{t}_j$, we have to set to zero the partial derivatives of the objective function J in (1) with respect to these parameters, using (7) as the distance measure:

$$\frac{\partial J}{\partial \boldsymbol{t}_j} = \sum_{i=1}^{N} (2u_{ij}^m)(R_j s_j \boldsymbol{p}_{ij}^* + \boldsymbol{t}_j - \boldsymbol{x}_i) = 0 , \tag{8}$$

$$\frac{\partial J}{\partial s_j} = \sum_{i=1}^{N} (2u_{ij}^m)(R_j s_j \boldsymbol{p}_{ij}^* + \boldsymbol{t}_j - \boldsymbol{x}_i)^T (R_j \boldsymbol{p}_{ij}^*) = 0 , \tag{9}$$

and

$$\frac{\partial J}{\partial \theta_j} = \sum_{i=1}^{N} (2u_{ij}^m) s_j (R_j s_j \boldsymbol{p}_{ij}^* + \boldsymbol{t}_j - \boldsymbol{x}_i)^T \left(\frac{dR_j}{d\theta_j} \boldsymbol{p}_{ij}^* \right) = 0 . \tag{10}$$

It is difficult to find analytical solutions that simultaneously satisfy (8)-(10). However, we can find closed-form expressions if we choose to update one parameter at a time. The following equations are obtained by solving (8) for $\boldsymbol{t}_j$, (9) for s_j, and (10) for θ_j, respectively:

$$\boldsymbol{t}_j = \frac{\sum_{i=1}^{N} u_{ij}^m (\boldsymbol{x}_i - R_j s_j \boldsymbol{p}_{ij}^*)}{\sum_{i=1}^{N} u_{ij}^m} , \tag{11}$$

$$s_j = \frac{\sum_{i=1}^{N} u_{ij}^m (\boldsymbol{x}_i - \boldsymbol{t}_j)^T (R_j \boldsymbol{p}_{ij}^*)}{\sum_{i=1}^{N} \left\| \boldsymbol{p}_{ij}^* \right\|^2 u_{ij}^m} , \tag{12}$$

and

$$\theta_j = \tan^{-1}\left[\frac{\sum_{i=1}^{N} u_{ij}^m (\boldsymbol{x}_i - \boldsymbol{t}_j)^T \, \boldsymbol{p}_{ij}^*}{\sum_{i=1}^{N} u_{ij}^m (\boldsymbol{x}_i - \boldsymbol{t}_j)^T \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \boldsymbol{p}_{ij}^*} \right] . \tag{13}$$

We can see that (11) reduces to the update equation for prototypes in standard fuzzy and possibilistic c-means clustering when we use point prototypes which, without loss of generality, mean $\boldsymbol{p}_{ij}$*=0 (i.e., only one vertex and no edge in the template).

A special consideration here is that (13) gives two values for θ_j over a range of 2π. We solve this problem by computing the objective function J using both values of θ_j for the prototype, and then choose the one that gives the lower value of J. Another consideration is the possibility of (12) to become negative. If this does occur, we set s_j to its absolute value and change θ_j by π.

Each of (11)-(13) can be used as a separate update step within one iteration of an alternating optimization algorithm. This actually allows for more flexibility to, say, "disable" a type of transform. For example, if we are looking for circles, we can simply skip (13) because we do not need to rotate the prototypes.

There is still one complication related to these update equations: Strictly speaking, the points $\boldsymbol{p}_{ij}$ themselves are also dependent on the prototype parameters. However, we are unable to express this dependence in a differentiable function. Our solution is to keep the points $\boldsymbol{p}_{ij}$ unchanged while updating the prototype parameters. One more step is then included in each iteration of alternating optimization to recalculate $\boldsymbol{p}_{ij}$ after the prototype parameters have been updated.

The resulting possibilistic c-template (PCT) clustering algorithm is listed below:

```
Initialize $s_j$, $\theta_j$, and $\boldsymbol{t}_j$ for the prototypes
Loop
  Find all $\boldsymbol{p}_{ij}$ and corresponding $\boldsymbol{p}_{ij}$*
  Compute $d_{ij}$ using (7)
  Update $u_{ij}$ using (2)
  Update $\boldsymbol{t}_j$ using (11)
  Update $s_j$ using (12)
  Update $\theta_j$ using (13)
Until convergence or the maximal allowed iterations
```

Convergence here is defined as when the change of each prototype between consecutive iterations is below a certain threshold. For this purpose, we define here a difference measure between two prototypes. Let $\{V_A, E_A\}$ and $\{V_B, E_B\}$ be the vertex and edge sets of the two prototypes, respectively. The difference measure is given by

$$\max\left[\max_{v \in V_A} \min_{e \in E_B} dist(v,e), \; \max_{v \in V_B} \min_{e \in E_A} dist(v,e) \right], \tag{14}$$

with *dist*(*v*,*e*) being the Euclidean distance between a vertex *v* and an edge *e*. Due to the nature of possibilistic clustering, convergence can be determined separately for each cluster.

2.3 A Robust Implementation of Possibilistic C-Template Clustering

There are two additional issues that are challenging for hard, fuzzy, and possibilistic c-shell clustering: The need to specify the number of clusters at the start, and sensitivity to prototype initialization. These challenges are also applicable to our c-template clustering algorithm. One advantage of the possibilistic approach is that it allows multiple prototypes to overlap with each other. As a result, we can just over-specify the number of clusters, and at the end just combine overlapping prototypes to obtain the correct number of clusters. However, possibilistic clustering is known to be extremely sensitive to good initialization. The solution proposed in [12] is to use fuzzy clustering first to obtain the initialization. In actual applications, such as in [9,10], initial prototypes are determined based on the distribution of edge pixels in images.

To simultaneously deal with both challenges, we implement a method that incorporates ideas from progressive circular shell clustering as described in [3], with modifications specifically for possibilistic clustering and to reduce the possibility that the clustering results are trapped in local minima of the objective function. This is because the problem of local minima is more profound with line-segment-based shells than with circles, as has been observed for rectangular clusters [7]. The following briefly describe techniques used in robust PCT clustering algorithm:

(1) The parameter η_j is updated in a coarse-to-fine manner. It is set to a relatively large value ($\eta_j^{1/2} \approx$ (total range of data)/5) for a newly initialized prototype to make sure that it is able to move toward nearby data. It is reduced by a multiplicative factor of 0.5–0.8 after each iteration until it is close to convergence, after which it is set according to the formula suggested in [12].

(2) The "goodness" of a cluster is determined by its density, a concept given in [3,5].

(3) Two prototypes that are very similar to each other according to (14) are merged.

(4) Prototypes with very low density at convergence are simply deleted.

(5) Prototypes with density above a threshold (at most one at each iteration) is considered a "good" one and moved to a separate list. Data points within a narrow window from this prototype are also removed.

(6) For a converging prototype of medium density, we "disturb" it, with some probability, by randomly modifying its parameters and/or increasing its η_j.

(7) Deleted, merged, or removed good prototypes are replaced with new randomly initialized prototypes.

(8) The main loop terminates after the total amount of remaining data is less than a pre-specified threshold (currently set at 5% of original data), or after no new "good" clusters are detected after a certain number of iterations.

(9) After the main loop terminates, we repeat the clustering process with the remaining prototypes plus the previously extracted "good" clusters as the initial prototypes and the original set of data points. Merging and deletion of bad clusters are performed only at the end of this step.

(10) Finally, we iteratively select and remove the remaining prototype with the highest density, remove its associated data points, and re-compute the density of the other prototypes. This is repeated until no prototypes with density above a pre-specified threshold remains. Prototypes selected in this stage are the final prototypes.

The steps (3)-(5) and (9) are based on ideas of progressive clustering in [3], with the others being our innovation.

3 Experiments

In this section we present our experiment results using various data sets. The first part includes synthetic data sets consisting of data points that form various shapes. The second part includes results using edge pixels that are extracted from real images. For each set of data points, we only look for clusters of one particular shape. We include synthetic data sets with and without noise points and minor scatters.

In Fig. 2 we display results using several synthetic data sets. Each pair contains results using the same templates but with noiseless and noisy data, respectively. Detected prototypes are plotted with the data. The results indicate that the algorithm is capable of detecting the desired shapes in the presence of scatter and noise, and of detecting the correct number of clusters.

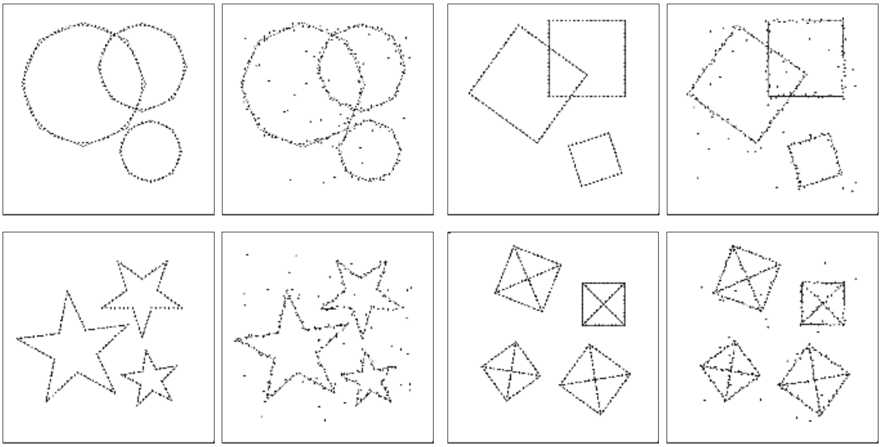


Fig. 2. Detection of four different shapes in synthetic data. Both noiseless and noisy data are presented here.

Fig. 3-5 display examples of detecting particular shapes in real images through possibilistic c-template clustering. All these data sets contain some scatter and noise. Fig. 3 shows the detection of circles of various sizes. Fig. 3(a)-(d) are the original image, the data points that are edge pixels, the data points with final prototypes overlapped with them, and the original image with final prototypes overlapped on it. All

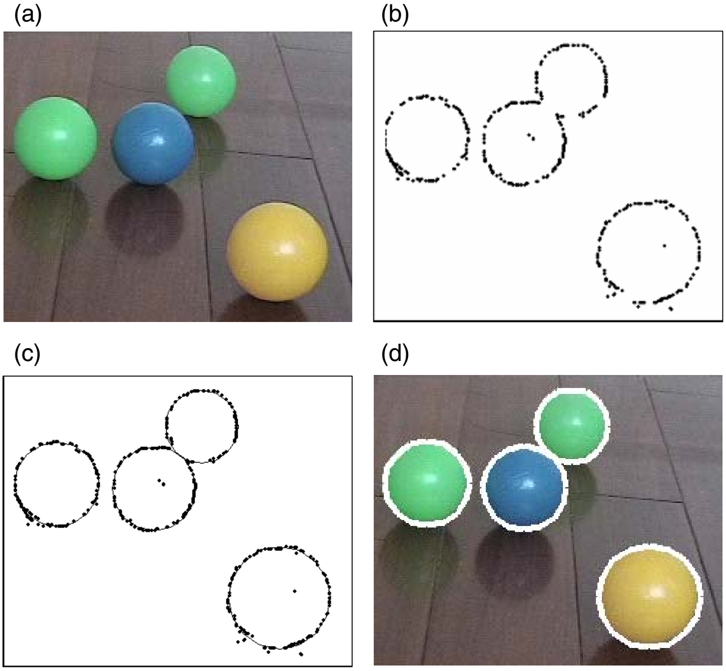


Fig. 3. Detection of circles in real-world images. (a) The original image. (b) The set of edge points used for clustering. (c) Same with (b), with final selected prototypes overlapped. (d) Same with (a), with detected circles overlapped.

the circles are detected correctly. While many existing algorithms can detect circular shapes, we include the results here to show that circles can be treated as a special case of templates as well. In addition, our algorithm is very efficient, usually converging with the correct number and location of circles in one pass within 20-30 iterations.

Fig. 4 contain example clustering results of more generic shapes. Each row in Fig. 4 corresponds to a different image and displays, from left to right, the original image, the edge pixels used for clustering, and the original image with the final prototypes overlapped as white lines. The last two templates in Fig. 1 are used with the first and second rows here, respectively. Since all the beads are of very similar sizes, we choose to skip the update of scaling. This is a reasonable condition for machine vision problems in controlled settings, such as in inspection applications. We consider this a demonstration of the flexibility of our template clustering algorithm, allowing the simplification based on information known *a priori*. Due to the increased complexity of these templates, these data sets typically take longer (50-100 iterations) to converge. We can also see that not all the prototypes are fitted to the data exactly. This is more evident with the double-ellipse prototypes in the second row. However, we still obtain the correct number and approximate locations of the objects of interest in each case.

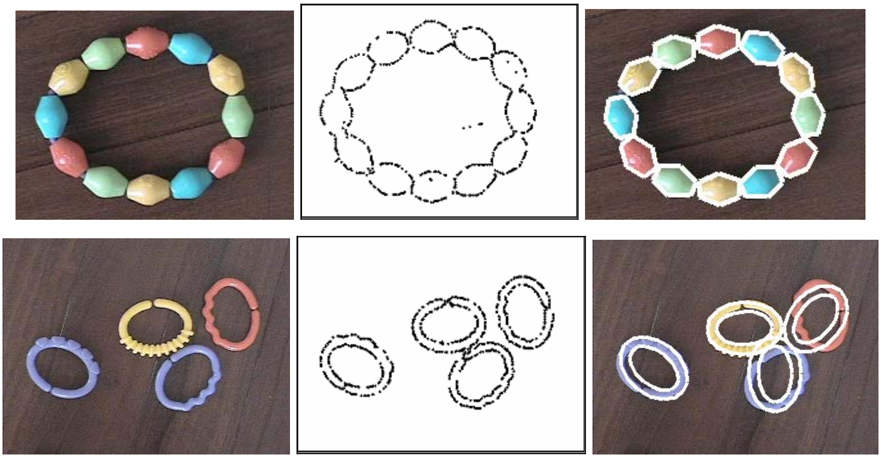


Fig. 4. Detection of generic shapes in real-world images. The two rows correspond to two images containing objects of two deferent shapes. Each row contains, from left to right, the original image, the set of edge points used for clustering, and the original image with final prototypes (detected objects) indicated by white lines.

4 Conclusions

This paper describes the algorithms and update equations that facilitate possibilistic c-shell clustering with generic-shaped template-based prototypes using the efficient alternating optimization scheme. The separation of prototype update equations allows us more flexibility in taking advantage of known properties of the clusters. The feasibility of this approach is illustrated with both synthetic data and real images. Techniques to make the process more robust with respect to number and initialization of prototypes are also discussed.

A number of research issues remained with this approach. For example, transforms more flexible than the current choices, such as a separate scaling factor for each dimension, should further expand the possible applications of this algorithm. More flexibility will also necessitate better robust shell clustering techniques. Another possibility is to combine GA with our AO algorithm to get the benefits of both methods, an approach that have been used in hard clustering with point-based prototypes [14,15].

Overall, we believe that further study and understanding of this technique will help expand the application of shell clustering more image analysis problems.

References

1. Anderson, I., Bezdek, J.C.: An Application of the C-Varieties Algorithm to Polygonal Curve Fitting. IEEE Trans. Sys. Man Cybernet., Vol. 15, (1985) 637-641
2. Dave, R.N.: Fuzzy Shell-Clustering and Application to Circle Detection in Digital Images. Int. J. Gen. Syst., Vol. 16 (1990) 343-355

3. Krishnapuram, R., Nasraoui, O., Frigui, H.: The Fuzzy C Spherical Shells Algorithm: A New Approach. IEEE Trans. Neural Networks, Vol. 3 (1992) 663-671
4. Krishnapuram, R., Frigui, H., Nasraoui, O.: Fuzzy and Possibilistic Shell Clustering Algorithms and Their Application to Boundary Detection and Surface Approximation - Part I. IEEE. Trans. Fuzzy Systems, Vol. 3 (1995) 29-43
5. Krishnapuram, R., Frigui, H., Nasraoui, O.: Fuzzy and Possibilistic Shell Clustering Algorithms and Their Application to Boundary Detection and Surface Approximation - Part II. IEEE. Trans. Fuzzy Systems, Vol. 3 (1995) 44-60
6. Frigui, H., Krishnapurum, R.: A Comparison of Fuzzy Shell Clustering Methods for the Detection of Ellipses. IEEE Trans. Fuzz. Sys., Vol. 4 (1996) 193-199
7. Hoeppner, F.: Fuzzy Shell Clustering Algorithms in Image Processing: Fuzzy C-Rectangular and 2-Rectangular Shells. IEEE Trans. Fuzzy Systems, Vol. 5 (1997) 599-613
8. Gao, X.-B., Xie, W.-X., Liu, J.-Z., Li, J.: Template Based Fuzzy C-Shells Clustering Algorithm and Its Fast Implementation. Proc. ICSP, Vol. 2 (1996) 1269-1272
9. Barni, M., Gualtieri, R.: A New Possibilistic Clustering Algorithm for Line Detection in Real World Imagery. Pattern Recognition, Vol. 32 (1999) 1897-1909
10. Barni, M., Mecocci, A., Perugini, L.: Craters Detection via Possibilistic Shell Clustering. Proc. IEEE Int'l Conf. Image Processing, Vol. 2 (2000) 720-723
11. Gath, I., Hoory, D.: Detection of Elliptic Shells Using Fuzzy Clustering: Application to MRI Images. Proc. ICPR, Vol. 2 (1994) 251-255
12. Krishnapuram, R., Keller, J.M.: A Possibilistic Approach to Clustering. IEEE Trans. Fuzz. Sys., Vol. 1 (1993) 98-110
13. Hall, L.O., Ozyurt, I.B., Bezdek, J.C.: Clustering with a Genetically Optimized Approach. IEEE Trans. Evolutionary Computation, Vol. 3 (1999) 103-112
14. Scheunders, P.: A Genetic C-Means Clustering Algorithm Applied to Color Image Quantization. Pattern Recognition, Vol. 30 (1997) 859-866
15. Sheng, W., Swift, S., Zhang, L., Liu, X.: A Weighted Sum Validity Function for Clustering with a Hybrid Niching Genetic Algorithm. IEEE Trans. Sys. Man. Cibernet. Part B, Vol. 35 (2005) 1156-1167

Position Estimation of Solid Balls from Handy Camera for Pool Supporting System

Hideaki Uchiyama and Hideo Saito

Keio University, 3-14-1 Hiyoshi, Kohoku-ku 223-8522, Japan

Abstract. This paper presents a method for estimating positions of solid balls from images which are captured using a handy camera moving around the pool table. Since the camera moves around by hand in this method, the motion of the camera in 3D space should be estimated. For the camera motion estimation, a homography is calculated by extracting the green felt region of the table-top area that is approximated to a polygon. Then, the balls are extracted from the table-top region for obtaining the positions of the balls. The 3D position of each ball is estimated using a projection matrix determined by the homography. The ball areas are classified by distribution of RGB data in each area. We apply our method to image sequences taken with a handy camera for evaluating the accuracy of the ball position estimation. By this experiment, we confirm that the accuracy of the estimated position is up to 18mm error, which is sufficiently small for displaying the strategy information in the pool supporting system.

1 Introduction

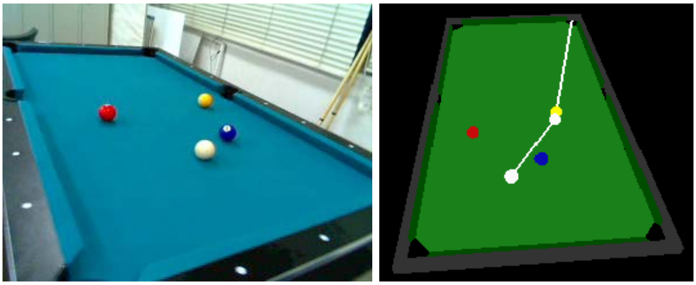
Pool is one of complex sports in the world that need knowledge of physical laws to decide direction and strength of a shot based on the arrangement of balls. However, it is difficult for beginners to shoot a ball considering dynamic behavior because they concentrate on shooting the center of the ball accurately. Since they can shoot the ball easily if the system teaches users the desired tracks of the ball, some pool supporting systems are proposed. Jebara used a head mounted live video display (HMD) with a camera and lines of a desired trajectory of the balls are rendered on HMD[1]. Also, Larsen used a computer controlled laser pointer to show a correct layout of the balls and the target[2][3][4]. However, such equipments are not desirable in pool game because playing pool with HMD is not natural for users, and installing the laser pointer system in a usual environment of pool hall is difficult. To be used popularly, the system needs to be composed of small equipments such as a cellular phone that have a camera, a small screen and CPU.

This paper describes a method for estimating the positions of solid balls from image sequence taken with a handy camera. The estimated positions are then used for showing supporting information on a LCD display. As for the balls, there are stripe balls and solid balls in pool game. In Chua's research, they classify two kinds of balls for Eight-Ball game which is one of the most famous pool games in the world[5]. Since we intend to apply our method to Nine-Ball game which is also a famous pool game, we propose the method for classification of colors of solid balls.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 393–402, 2006.

A Ball become a circle and a table becomes a rectangle as a 2D objects when an image is captured from table top[5][6][7][8]. However, we treat balls and a table as a 3D object since it is difficult for users to capture images from the table-top. In our method, we assume that images as shown in Fig.1 (a) are captured with a handy camera by moving around the pool table. To each image, we extract a green felt area to compute a homograpy by using planarity of the area. After extracting the area, we extract circular regions as balls and then determine the center positions of the circles. 3D coordinates of the centers of balls can be estimated using a projection matrix computed by the homograpy of the table-top plane. To avoid accidental miss-detection of the balls, we use multiple frames in the input image sequence. Then users judge the correctness of the estimated 3D positions by watching the arrangement of the balls.

We apply our method to image sequences taken with a handy camera for evaluating the accuracy of the estimated ball positions. By this experiment, we confirm that the accuracy of the estimated position is up to 18mm error, which is sufficiently small for the pool supporting system.



(a) Input (b) Desired output

Fig. 1. Input and output

2 Proposed Method

In Jebara's research[1], captured images from eye position were divided into a green felt region, balls, corners, and pockets by using a probabilistic color model for understanding the status of the table-top. The white lines which connected a target ball with a desired pocket to provide strategic information to players. Because they don't compute the geometrical relationships of all the balls and others, they can display only simple lines. In our research, we compute the 3D positions of all the balls on the table by capturing images of the whole parts of the table for pool supporting system based on the arrangement of all the balls.

2.1 Overview

Input images are of the whole of the pool table captured from different viewpoints, and then our method is applied to each image.

First, a green felt region of the table is extracted by color segmentation to compute a homography. Then, the region is approximated to a quadrangle because four vertices are necessary for computing the homography. After computing the homography, a projection matrix is computed from the homography by Simon's method[9][10][11].

Next, non-green areas in the green felt area are extracted as candidates of balls. Pockets and cushions are included in the candidates. Balls are extracted by the size of an area and the degree of circularity in each candidate. For each ball, the number of the ball is determined by distribution of RGB data.

In the end, 3D positions of the balls are computed by using the projection matrix and the 2D coordinates of the centers of the balls in images. To decrease false detections, a user determines whether the result is sufficient or not.

In our research, we implement the proposed algorithm by using OPENCV[12], and introduce the functions which we use.

2.2 Green Felt Area Detection

It is necessary for computing a homography to detect a plane. Both a frame and the inside of the pool table which are made of a green felt are planes shown in Fig. 2(a). Because of cushions, the whole part of the inside cannot be detected like the parts of the red circle of Fig. 2(a). All regions of the green felt that are a frame and the inside are extracted in our method.

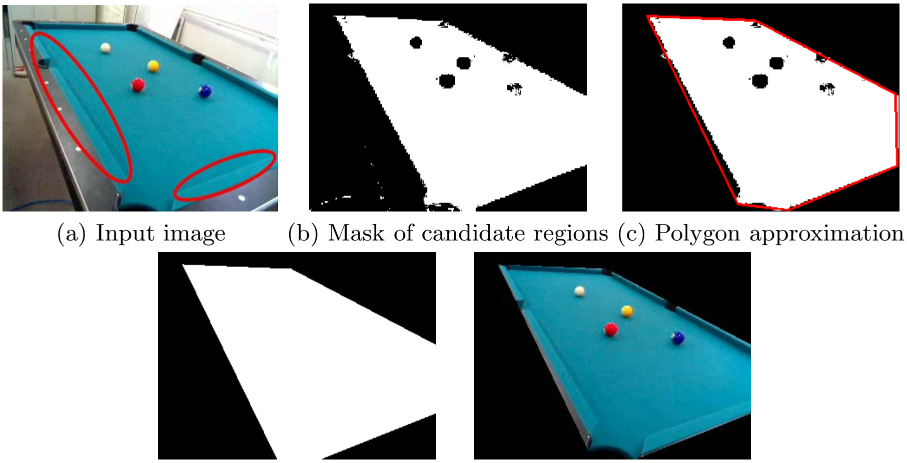
To extract the green felt area by color segmentation, RGB vector of the area is measured beforehand as a template. By computing the angle between RGB vector of each pixel and that of the template, each pixel is determined whether included in the area or not. Each pixel is included in the area when the angle is below a threshold because the angle is small if the colors of two vectors are close. Fig. 2(b) depicts the mask of candidates of the area.

Next, the largest region in the mask is extracted, and approximated to a polygon as shown in Fig. 2(c) using Approxpoly() of the OPENCV function. From the sides of the polygon, four sides from longer ones are chosen, and the quadrangle is made by these sides as the mask of the green felt area in Fig. 2(d). Fig. 2(e) shows the green felt area which is extracted from a input image segmented by the mask.

2.3 Camera Calibration

The green felt area is extracted and approximated to a quadrangle in Section. 2.2 to compute a homography $\boldsymbol{H}$. The homography is computed by associating the vertices of the green felt area of an actual pool table and those of the quadrangle in a input image. To compute a projection matrix by the homography, we employ Simon's method.

A 3D coordinate system is related to a 2D coordinate system by 3×4 projection matrix $\boldsymbol{P}$. Thus, each 3D coordinate system designed for each plane is also related to the input images by each projection matrix. If a Z coordinate of each plane is set to 0, the homography $\boldsymbol{H}$ also relates between each plane and the input images. In Eq. (2), the projection matrix $\boldsymbol{P}$ is composed of intrinsic



(a) Input image (b) Mask of candidate regions (c) Polygon approximation

(d) Mask of a green felt area (e) Extracted green felt area

Fig. 2. Green felt area detection

parameters $\boldsymbol{A}$, a rotation $\boldsymbol{R}$ and a translation $\boldsymbol{t}$ of extrinsic parameters. Also, the homography $\boldsymbol{H}$ is expressed in Eq. (3) which is the deleted $\boldsymbol{r}_3$ of $\boldsymbol{P}$.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \simeq \boldsymbol{P} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \simeq \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \simeq \boldsymbol{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{1}$$

$$\boldsymbol{P} = \boldsymbol{A}\,[\boldsymbol{R} \mid \boldsymbol{t}] = \boldsymbol{A}\,[\boldsymbol{r}_1 \boldsymbol{r}_2 \boldsymbol{r}_3 \boldsymbol{t}] \tag{2}$$

$$\boldsymbol{H} = \boldsymbol{A}\,[\boldsymbol{r}_1 \boldsymbol{r}_2 \boldsymbol{t}] = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{3}$$

First, intrinsic parameters $\boldsymbol{A}$ are computed by the homograpy $\boldsymbol{H}$. We assume that the skew is 0, the aspect ratio is 1 and the principal point is the center of the image. The intrinsic parameters can be defined as in Eq. (4). Then, we only have to estimate the focal length f. Using the property of the rotation matrix $\boldsymbol{R}$ that is the inner product of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ is equal to 0, the focal length f is computed as shown in Eq. (5).

$$\boldsymbol{A} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{pmatrix} (c_x, c_y) & : & \text{principal point} \\ f & : & \text{focal length} \end{pmatrix} \tag{4}$$

$$f = \frac{(h_{11} - c_x h_{31})(h_{12} - c_x h_{32}) + (h_{21} - c_y h_{31})(h_{22} - c_y h_{32})}{-h_{31} h_{32}} \tag{5}$$

Next, the rotation $\boldsymbol{R}$ and the translation $\boldsymbol{t}$ of extrinsic parameters are computed. $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and $\boldsymbol{t}$ are computed by the homography $\boldsymbol{H}$ and a inverse matrix of intrinsic parameters $\boldsymbol{A}$ shown in Eq. (6). $\boldsymbol{r}_3$ is computed by the nature of the rotatin matrix that the cross product of $\boldsymbol{r}_1$ and $\boldsymbol{r}_r$ becomes $\boldsymbol{r}_3$ in Eq. 7.

$$[\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{r}_3\boldsymbol{t}] = \boldsymbol{A}^{-1}\boldsymbol{H} \tag{6}$$

$$\boldsymbol{R} = \left[\boldsymbol{r}_1\ \boldsymbol{r}_2\ (\boldsymbol{r}_1 \times \boldsymbol{r}_2)\right] \tag{7}$$

2.4 Ball Detection

Balls are detected in the green felt area because the balls surely exist on the area. To detect the balls, non-green areas are extracted in the green felt area as candidates of the balls, and then balls are detected from candidates.

To extract the candidates of balls by color segmentation, RGB vector of the green felt used in Section. 2.2 is used as a template. For each pixel, the angle between RGB vector of each pixel and that of the template is computed. Also, the difference of Norm of two vectors that shows the difference of the brightness is calculated to determine whether the green felt is or not because there is the similar color of the felt in the colors of the balls. Each pixel is determined as a candidate of balls when the difference of the norm is over a threshold and the angle of two vectors is over another threshold. Fig. 3(b) shows candidates of the balls. Each candidate is determined whether it is the pixel of the balls or not by the size and the degree of circularity. The mask of the balls is made as shown in Fig. 3(d). Fig 3(e) depicts the balls which segmented by the mask from a input image.

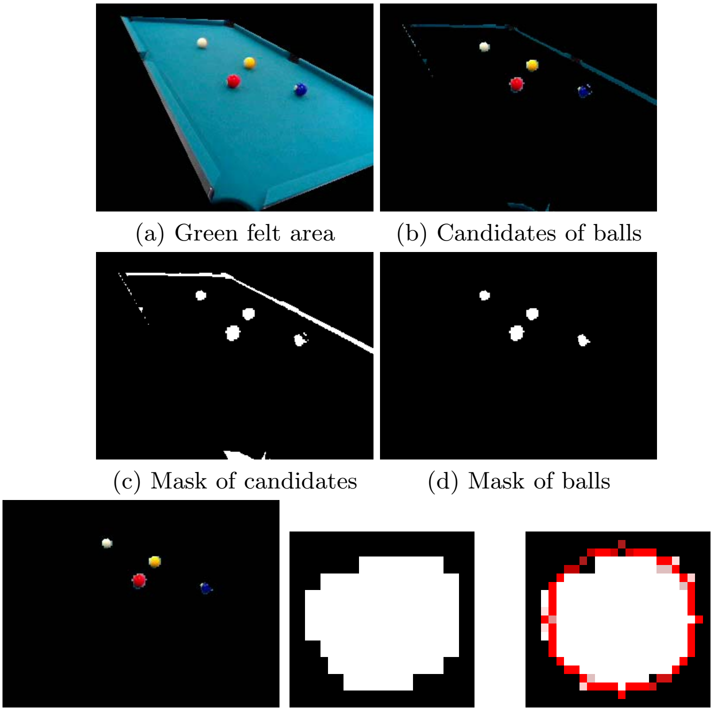
A ball in 3D world becomes a circle in 2D image. Then, each area of the balls in Fig. 3 is approximated by the circle in Fig. 3(f) by using cvMinEnclosingCircle() of the OPENCV function and the center coordinate of the circle is also calculated by the function to compute the 3D coordinates of the ball.

2.5 Classification of Solid Balls

We propose the classification of solid balls which colors are nine.

In the ball area depicted in Fig. 4, the shadow of the ball and the specular reflection element are also included. If the average of the color in the area is computed to classify the ball, miss-classification often occur because of the influence of the shadow and the specular reflection element strongly. To classify the balls precisely, we apply a voting such that the closest color of the balls is selected.

RGB vector of each ball is measured beforehand as a template. The angle between RGB vector of each pixel in the ball area and that of the template of each ball is calculated, and votes on the balls to which the angle is minimum like Table. 2.5 are done. Number zero is a white cue ball. Then, the ball with lion's share of votes is selected.



(a) Green felt area (b) Candidates of balls

(c) Mask of candidates (d) Mask of balls

(e) Balls (f) Mask of a ball (g) Circle approximation

Fig. 3. Detection of balls

2.6 Three Dimensional Coordinate of the Balls

In Eq. (8), (x, y) is 2D coordinates in the images and (X, Y, Z) is 3D coordinates in the world coordinate system. The projection matrix $\boldsymbol{P}$ is calculated by the homography in Section. 2.3.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \simeq \boldsymbol{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{8}$$

It is necessary to compute (X, Y, Z) to determine 3D coordinates of the center of the ball. However, it is not possible to compute from (x, y) in the image and the projection matrix $\boldsymbol{P}$ because the degree of freedom of 3D coordinates is three. Then, Z is calculated beforehand by nature of a sphere to decrease the degree of freedom.

As Fig. 5 is shown, the straight line that connects the point p in the image and the point$P$ in the world which appear in the image passes the center$C$ of the ball by nature of a sphere. X and Y can be computed because the degree of freedom becomes two if Z is known. Since the plane which used when the

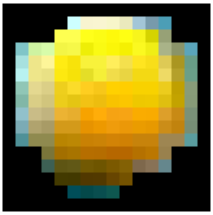


Fig. 4. Distribution of color

Table 1. Judgement of balls by vote

ball number	0	1	2	3	4	5	6	7	8	9
result of vote	23	83	0	0	0	0	32	0	15	0

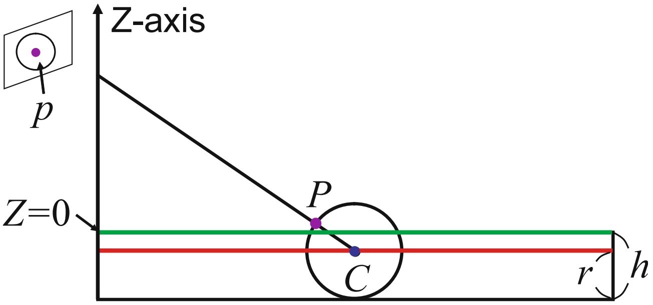


Fig. 5. Computation of the three dimentional position of a ball center

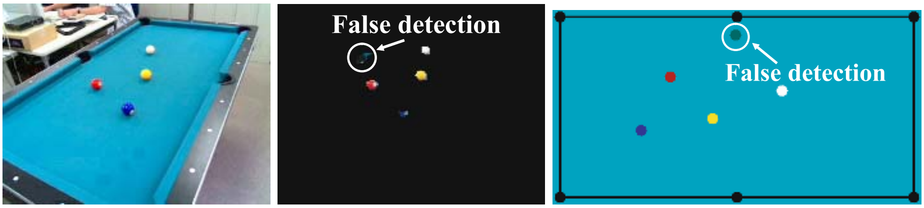
homography is calculated is $Z = 0$, Z of the center of the balls is determined as $Z = -(h - r)$ because the radius r of the ball and the height h of the cushion of a pool table are known. X and Y can be computed by using $\boldsymbol{P}$, (x, y), and Z.

2.7 Judgment of the Result by User

When the 3D positions of the balls are estimated in each image, the pocket or parts of the pool table might be mistakenly detected as a ball in Fig. 6(b). To decrease such false detections, a user determines whether the estimated positions of the balls are correct or not by watching the display of the arrangement of the balls from the top of the table as shown in Fig. 6(c) when the positions can be computed for the first time. This is repeated until the user satisfies the estimated arrangement of the balls. As for the estimated positions in the image taken from other aspects, correctness is automatically judged depending on the distance with the position of the ball that has already been computed. Then, the average of the computed positions is calculated. In the way, the false detections can be removed by user interaction.

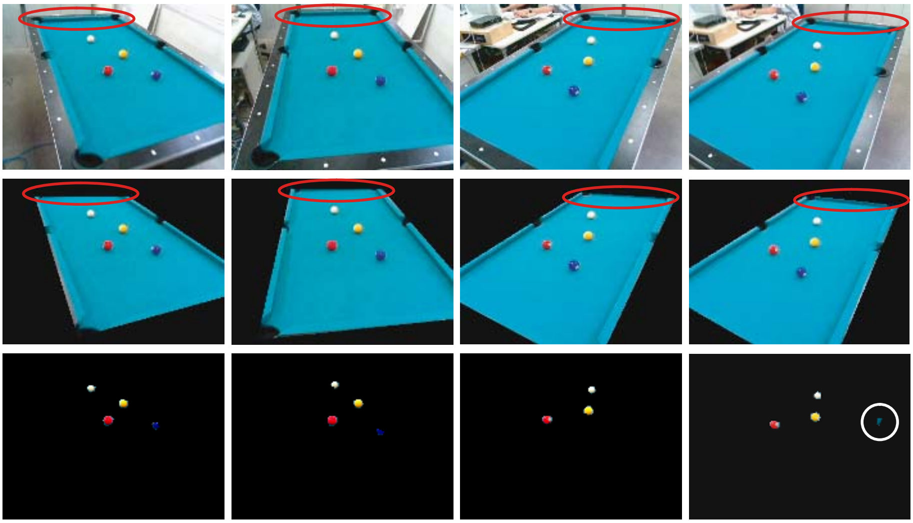
3 Experiments

We apply our method to some image sequences including 150 images taken with a handy camera for evaluating the accuracy of the estimated 3D ball position



(a) Input (b) False detection (c) Display for users

Fig. 6. Judgment of the result by user



(a) frame 20 (b) frame 50 (c) frame 80 (d) frame 90

Fig. 7. Results

estimation. As Z coodinate is constant depicted in Section. 2.6, we confirm the accuracy of (X, Y).

The size of table is 1330mm×700mm and that of images is 320 × 240pixels. We used four balls which colors are white, yellow, blue and red. We put on the balls which coordinate is measured with a tape measure beforehand, and then we compare their coordinates with the coordinates estimated from the input images. Fig. 7 show results that the first row shows inputs, the second row shows grenn felt areas, and the third row shows balls.

First, we count the number that balls are extracted from 140 images. The result is shown in Table. 2. Since we captured images freehand, 34 images blurred and all part of a board was not include in these images . We couldn't extract a board from such images, and alse balls. White, yellow and red balls can be extracted in many images because their colors are quite different from the color of the green felt. On the other hand, a blue ball cannot sometimes be extracted

Table 2. Number of balls extracted

white	yellow	blue	red	corner
103	116	82	116	5

because its color is a little close to the color of green felt. Pocket is also extracted as a ball shown in a ball area image of Fig. 7(d) because the shape of pocket is close to the circle and the color is different from that of the green felt.

Next, we compare the ball positions measured with a tape measure with the estimated positions from the input images via the proposed method. Table. 3 shows the result of comparison. The result shows that Y values contain more errors than Xvalue. The error is mostly caused by the unstable detection of the far side edge of the green felt region. As shown in Fig. 7, the far side edge indicated by the red ellipse cannot clearly be captured in the input image. In Fig. 8, square dots show ground truth positions, while circular dots show the estimated positions from the input images. Even though there are around 18mm errors in Y components, the arrangement of the balls estimated via the proposed method is similar to the ground truth, therefore we consider that the proposed method can sufficiently be used for the pool supporting system.

Next, the center position of the white ball computed in each frame are shown in Fig. 9. The error in each frame is not constant because the lighting condition is different, depending on the direction where a pool table is captured.

Table 3. Ball positions(m)

ball color	white		yellow		blue		red	
coordinate	x	y	x	y	x	y	x	y
ground truth	0.373	0.823	0.273	0.573	0.223	0.323	0.423	0.423
estimated from images	0.378	0.841	0.274	0.582	0.228	0.317	0.429	0.425

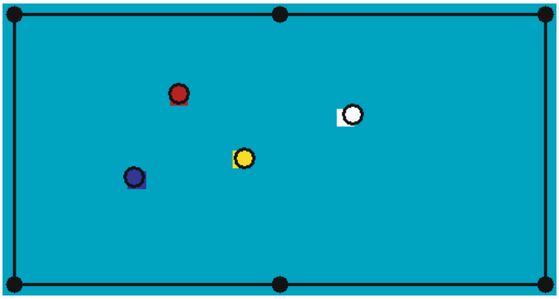


Fig. 8. Comparison of ground truth (rectangular dots) with estimated from image (circular dots)

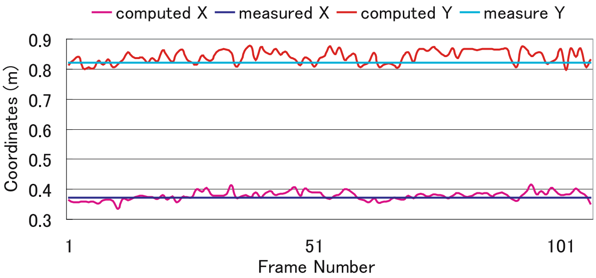


Fig. 9. Coordinates of white ball

4 Conclusions

We have proposed a method for estimating positions of solid balls from images which are captured using a handy camera moving around the pool table. For estimating the positions of all the balls on the pool table, this method first computes homographies of the table-top region by extracting the green felt area via color segmentation. The computed homographies provide camera parameters so that the 3D positions of the balls can be estimated from the extracted positions of the balls in the input images. In the result, we have shown that the position estimation accuracy is up to around 18 mm error, which is sufficiently accurate for the pool supporting systems.

References

1. T.Jebara, et al.: Stochastics: Augmenting the Billiards Experience with Probabilistic Vision and Wearable computers, Proc. of ISWC, pp. 138-145, 1997.
2. L.B.Larsen, et al.: The Automated Pool Trainer - A multi modal system for learning the game of Pool, Proc. of ICIMADE, pp. 90-96, 2001.
3. L.B.Larsen and T.Brondsted: A Multi Modal Pool Trainer, Proc. of IPNMD, pp. 107-111, 2001.
4. L.B.Larsen, M.D.Jensen and W.K.Vodzi: Multi Modal User Interaction in an Automatic Pool Trainer, 4th IEEE ICMI, pp. 361-366, 2002.
5. S.C.Chua, E.K.Wong and V.C.Koo: Pool Balls Identification and Calibration for a Pool Robot, Int. Conf. on ROVISP, pp. 312-315, 2003.
6. S.C.Chua, E.K.Wong and V.C.Koo: Simulation of A Robotic Pool, M2USIC, 2004.
7. S.C.Chua, et al.: Decision Algorithm for Pool Using Fuzzy System, Proc. of iCAiET, pp. 370-375, 2002.
8. M.E.Alian, et al.: Roboshark: a Gantry Pool Player Robot, 35th ISR, 2004.
9. G.Simon, A.W.Fitzgibbob and A.Zisserman: Markerless tracking using planar structures in the scene, Proc. of the ISAR, pp. 120-128, 2000.
10. G.Simon and M.Berger: Reconstructing while registering: a novel approach for markerless augmented reality, In Proc. of the ISMAR, pp. 285-294, 2002.
11. Y.Uematsu and H.Saito: Vision-based Registration for Augmented Reality with Integration of Arbitrary Multiple Planes, 13th ICIAP, pp. 155-162, 2005.
12. Intel: http://www.intel.com/

Shape Retrieval Using Statistical Chord-Length Features

Chaojian Shi[1,2] and Bin Wang[2,⋆]

[1] Merchant Marine College, Shanghai Maritime University,
Shanghai, 200135, P.R. China
[2] Department of Computer Science and Engineering, Fudan University,
Shanghai, 200433, P.R. China
cjshi@shmtu.edu.cn, wangbin.cs@fudan.edu.cn

Abstract. A novel shape description method, statistical chord-length features (SCLF), is proposed for shape retrieval. SCLF first describes the contour of a 2D shape using $k/2$ one-dimensional chord-length functions derived from partitioning the contour into k arcs of the same length, where k is the parameter of SCLF. The means and variances of all the chord-length functions are then calculated and a k dimensional feature vector is generated as a shape descriptor. Two experiments are conducted and the results show that SCLF achieves higher retrieval performance than traditional description methods such as geometric moment invariants and Fourier descriptors.

1 Introduction

With the rapid development of digital and informational technologies, more and more multimedia information is generated and available in digital form from varieties of sources all over the world. Most of the multimedia information is in the form of images. There is an urgent need for efficiently managing, organizing and navigating through them. Content-based image retrieval (CBIR)[1] is a research area dedicated to address this issue. CBIR utilizes low-level image features such as color, texture and shape to search through image databases. Due to its potential applications, CBIR has attracted a great amount of attention in recent years [2,3,4].

Shape is a very important feature to human perception. Human beings tend to perceive scenes as being composed of individual objects, which can be best identified by their shapes. Therefore, shape based image retrieval (shape retrieval) is a primal element of CBIR. Basically, shape retrieval is the measuring of similarity between shapes represented by their features. Accordingly, how to describe the shape, i.e, how to extract the features of the shape is a key to shape retrieval. MPEG-7 sets six principles for a good shape descriptor, they are: (1)good retrieval accuracy, (2)compact features, (3)general application, (4)low computation complexity, (5)robust retrieval performance, and (6) hierarchical

[⋆] Corresponding author.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 403–410, 2006.

coarse to fine representation. In general, the existing shape description methods can be classified into two categories: contour-based methods and region-based methods. Region-based methods extract features from all the pixels within a shape, while contour-based methods only exploit shape boundary information.

The approach of geometric moment invariants proposed by Hu [5] is one of the region-based methods and have been used in many applications [6,7,8]. This method is based on the work of the 19th century mathematicians Boole, Cayley and Sylvester, and on the theory of algebraic forms,

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y), \; p, q = 0, 1, 2, \dots \; .$$

By using nonlinear combinations of the lower moments, a set of moment invariants are derived. The desirable properties of this descriptor is that it is invariant to translation, scaling and rotation. The main problem of geometric moment invariants is that a few invariants derived from lower moments only is insufficient to describe a shape accurately, while higher order invariants are difficult to derive.

Another classical shape description method is using Fourier descriptors. In this method, the contour of a 2D shape is first described by an 1-D function (termed contour function) and Fourier transform is then applied on the function. The normalized Fourier transform coefficients are then taken as shape descriptor. The complex coordinates and the cumulative angle function are dominantly used to derive FDs. Different contour functions will result in different FDs. Through comparing six existing contour functions, Zhang et al.[9] reported that for general shapes, the centroid distance function is the most desirable contour function to derive FD for shape retrieval. The advantage of FDs is that it is robust, compact and simple to compute. Zhang et al.[9] also found that 10 FD features are sufficient to describe a shape.

In this paper, we propose a novel shape descriptor, statistical chord-length features (SCLF), for shape retrieval. SCLF first describes the contour of a 2D shape using $k/2$ one-dimensional chord-length functions which are obtained by partitioning the contour into k arcs of the same length, where k is the parameter of SCLF. The means and variances of all the chord-length functions are then calculated and form a k dimensional feature vector, which is considered as a shape descriptor. SCLF is compact and simple to compute. It can characterize the shape more accurately than the above mentioned methods. Two experiments are conducted and the results show that SCLF achieves higher retrieval performance than geometric moment invariants and Fourier descriptors.

2 Statistical Chord-Length Features (SCLF)

2.1 Chord-Length Function

The contour C of a 2D shape can be denoted as an ordered sequence of N coordinate points, $C = \{\lambda_t = (x(t), y(t)), t = 0, 1, \dots, N-1\}$, where C is closed, i.e. $\lambda_{i+N} = \lambda_i$. The diameter D of the contour C is defined as

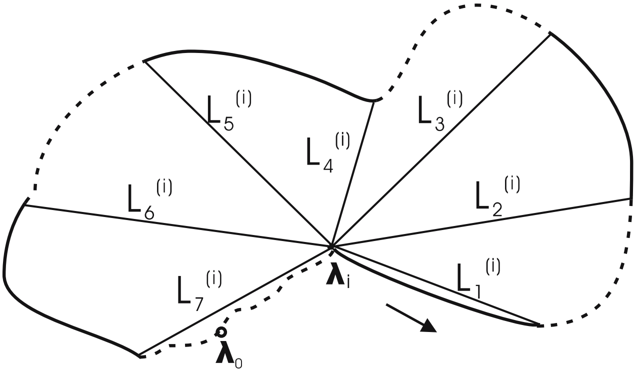


Fig. 1. An example of partitioning an contour into eight equal-arc-length sections, where λ_i is the starting point and $L_1, L_2, \ldots, L_7$ are the obtained chord lengths. And λ_0 is the reference point from which the contour is parameterized.

$$D = \max_{0 \le i,j \le N-1} \sqrt{(x(i) - x(j))^2 + (y(i) - y(j))^2}. \tag{1}$$

Let us take a point, $\lambda_i \in C$, as starting point and traverse the contour anti-clockwise to divide it into k sections $\widehat{\lambda_i S_1}, \widehat{S_1 S_2}, \ldots, \widehat{S_{k-1}\lambda_i}$ of equal arc length and obtain $k-1$ chords $\overline{\lambda_i S_1}, \overline{\lambda_i S_2}, \ldots, \overline{\lambda_i S_{k-1}}$, where S_j is the jth division point and $k > 1$ is a pre-specified parameter. We now have $k-1$ chord lengths $L_1^{(i)}, L_2^{(i)}, \ldots, L_{k-1}^{(i)}$, where $L_j^{(i)}$ is the length of the chord $\overline{\lambda_i S_j}$ which is defined as the Euclidean distance between the points, λ_i and S_j. Fig. 1 gives an example of partitioning a contour into eight equal-arc-length sections.

As the starting point, λ_i, moves along the contour, the chord lengths $L_j^{(i)}, j = 1, \ldots, k-1$, vary accordingly. In other words, $L_j^{(i)}$ is function of λ_i. Here we term it chord-length function and use L_j to denote it. Thereby $k-1$ chord-length functions, $L_1, L_2, \ldots, L_{k-1}$, are then obtained. The basic requirement for a shape descriptor is that it must be invariant to rotation, scaling and translation. Since the chord-length functions are derived from equally partitioning the contour and from moving the starting point, λ_i, along the contour, they are invariant to translation and rotation. However, the chord length is variant to spatial scaling. To solve this problem, each chord-length functions is normalized using the max chord-length, i.e. the diameter D of the contour. All the chord-length functions are then scaled to span the same range of values, say, $[0, 1]$. From the definition of the chord-length functions, since they are obtained by equally partitioning the contour, it is not difficult to find that only half of them are required, i.e. only $k/2$ chord-length functions, $L_1, L_2, \ldots, L_{k/2}$, are required for characterizing the shape. The advantage of using chord-length functions is that different level chord-lengths can capture both the global and local features of a shape. The chord length functions of the division points closer to the starting point, λ_i, are likely to capture the local features, and those of farther points can capture global features.

2.2 Statistical Feature Vector

Now, we have obtained $k/2$ chord-length functions which are invariant to rotation, scaling and translation. However, like other contour functions, they are not compact. Another problem is that they depend on the reference point, λ_0 (Fig. 1), from which the contour is parameterized. The reason of their dependence on the reference point is that the contour is closed and arbitrary point of the contour can be selected as reference point and the chord-length functions will accordingly be changed. For solving these problems, we calculate the mean m_j and variance σ_j of chord-length function $L_j,\ j = 1, 2, \ldots N-1$, as

$$m_j = \frac{1}{N} \sum_{i=0}^{N-1} L_j^{(i)} \tag{2}$$

and

$$\sigma_j = \frac{1}{N-1} \sum_{i=0}^{N-1} (L_j^{(i)} - m_j)^2. \tag{3}$$

A statistical feature vector $V = (m_1, \sigma_1, m_2, \sigma_2, \ldots, m_{k/2}, \sigma_{k/2})$ is then obtained and used as a shape descriptor.

3 Dissimilarity Measure

Here, we use the statistical feature vector to measure the dissimilarity of two shapes. Assume that $V^{(A)} = (m_1^{(A)}, \sigma_1^{(A)}, m_2^{(A)}, \sigma_2^{(A)}, \ldots, m_{k/2}^{(A)}, \sigma_{k/2}^{(A)})$ and $V^{(B)} = (m_1^{(B)}, \sigma_1^{(B)}, m_2^{(B)}, \sigma_2^{(B)}, \ldots, m_{k/2}^{(B)}, \sigma_{k/2}^{(B)})$ are statistical feature vectors of shape A and shape B, respectively. We use χ^2 statistics [10] to measure the distance between vectors $V^{(A)}$ and $V^{(B)}$ as follows

$$d_{\chi^2}(A, B) = \sum_{i=1}^{k/2} (\frac{(m_i^{(A)} - \overline{m_i})^2}{\overline{m_i}} + \frac{(\sigma_i^{(A)} - \overline{\sigma_i})^2}{\overline{\sigma_i}}) \tag{4}$$

where $\overline{m_i} = (m_i^A + m_i^B)/2$ and $\overline{\sigma_i} = (\sigma_i^A + \sigma_i^B)/2$.

4 Experimental Results and Discussions

To test the performance of the proposed SCLF, we conducted retrieval tests on two shape databases. One, as shown in Fig. 2, is a benchmark used in [11] and [12]. It includes nine categories and eleven instances are included in each of them, resulting in a total of 99 shape instances. Another, as shown in Fig. 3, is a database of leaf shapes which are taken from nature. It consists of 6 categories and 15 instances are included in each categories.

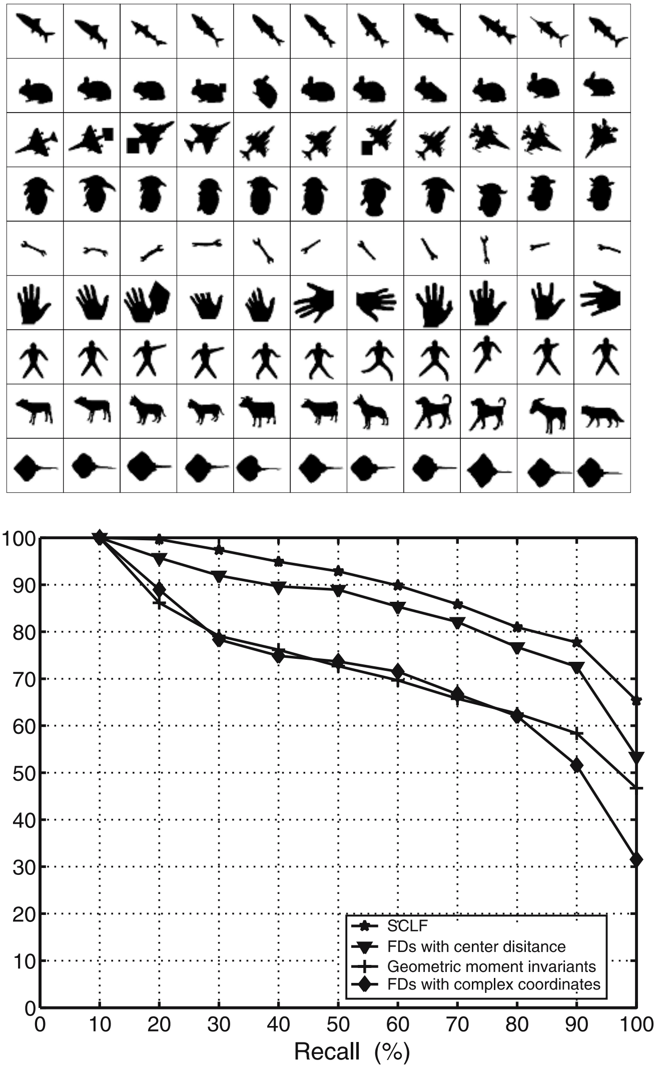


Fig. 2. Top: A database including 99 shapes [11]. Bottom: The *precision-recall plots* for SCLF, geometric moment invariants, FDs derived from centroid distance function and FDs derived from complex coordinates function.

Common performance measure, precision and recall of the retrieval [13], was used as the evaluation of the query results. Precision P is defined as the ratio of the number of retrieved relevant shapes r to the total number of retrieved shapes n, i.e, $P = r/n$. Recall R is defined as the ratio of the number of retrieved relevant shapes r to the total number m of relevant shapes in the whole database, i.e., $R = r/m$. Each shape in the database is used as a query. For each query, the precision of the retrieval at each level of the recall is obtained. The final precision of retrieval is the average precision of all the query retrievals.

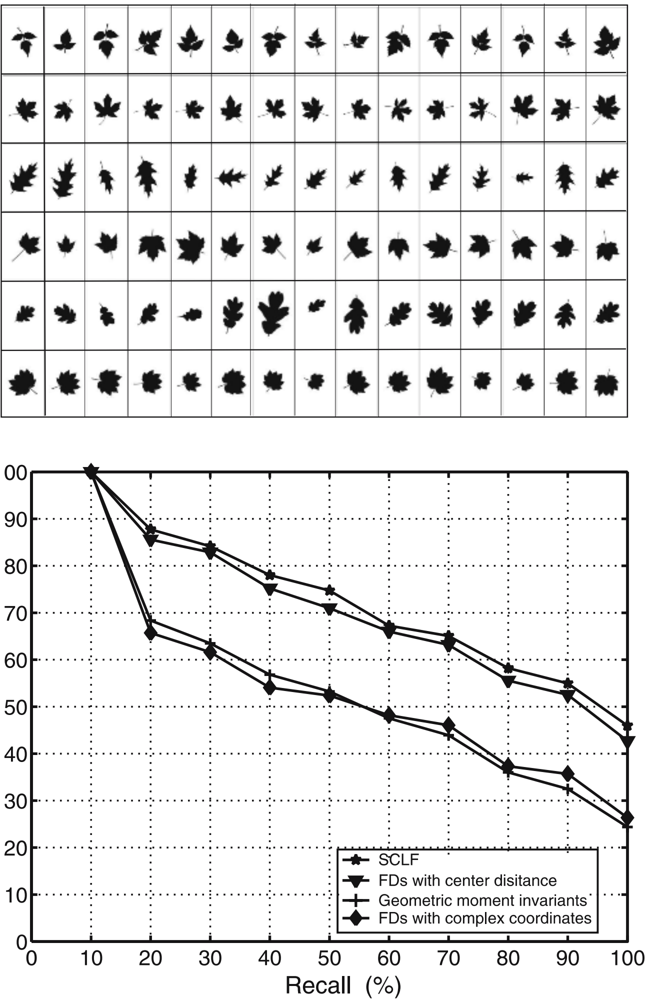


Fig. 3. Top: A database including 90 leaf shapes [11]. Bottom: The *precision-recall plots* for SCLF, geometric moment invariants, FDs derived from centroid distance function and FDs derived from complex coordinates function.

As for the parameter k, SCLF with small values of k tends to lose local features, and that with large k will increase computational complexity and be sensitive to noises. Our experiments indicate that for most of shapes, six to ten are reasonable values for k. We set the parameter k of SCLF to 8, i.e. we equally partitioned the contour into 8 segments. That is to say that the number of the elements of the obtained statistical feature vector is eight. To show the superiority of SCLF, three widely used shape description methods, geometric moment invariants, FDs derived from centroid distance function and FDs derived from complex coordinates function, were taken as comparisons. We performed the task of shape retrieval on the two shape databases using the four shape description methods, respectively. The obtained precision and recall plots for them are shown in Fig.2 and Fig. 3, respectively.

From the experimental results, we can see that the proposed SCLF achieves higher precisions at each level of recall than the three other shape description methods on both of the shape databases.

5 Conclusions

We have proposed a novel shape descriptor, statistical chord-length features (SCLF), for shape retrieval. By equally partitioning the contour, different level chord lengths can be obtained to capture both global and local features of a shape. Consequently, SCLF can describe the shape more accurately. Two experiments have been conducted to test the performance of SCLF. The results show that SCLF outperforms the traditional shape descriptors such as geometric moment invariants and Fourier descriptors.

Acknowledgement

The research work in this paper is partially sponsored by Shanghai Leading Academic Discipline Project, T0603

References

1. T. Kato, "Database Architecture for Content-Based Image Retrieval," *In image storage and retrieval systems*, Proc SPIE 1662, 112-123, (1992).
2. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos and G. Taubin, "The QBIC Project: Querying Image By Content Using Color, Texture and Shape," *In Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1908, 173-187, (1993).
3. Y. A. Aslandogan and C. T. Yu, "Techniques and Systems for Image and Video Retrieval," *IEEE Trans. On Knowledge and Data Engeering*, 11(1), 56-63, (1999).
4. A. Yoshitaka and T. lchikawa, "A Survery on Content-based Retrieval for Multimedia Databases," *IEEE Trans. On Knowledge and Data Engineering*, 11(1), 81-93, (1999).
5. M. K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Transactions on Information Theory*, IT-8; 179-187, (1962).
6. A.S. Dudani, K.J. Breeding, R.B. McGhee, "Aircraft identification by moment invariants," *IEEE Trans. Comput.*, C-26(1), 39-46, (1977).
7. S.O. Belkasim, M. Shridhar, M. Ahamdi, "Pattern recognition with moment invariants: a comparative study and new results," *Pattern Recognition*, 24(12), 1117-1138, (1991).
8. B.M. Mehtre, M.S. Kankanhalli, W.F. Lee, "Shape measures for content based image retrieval: a comparision," *Inf. Process. Manage.*, 33(3), 319-337, (1997).
9. D. Zhang, G. Lu, "Study and Evaluation of Different Fourier Methods for Image Retrieval," *Image and Vision Computing*, 23, 33-49, (2005).

10. Y. Rubner, "Perceptual Metrics for Image Databased Navigation," *PhD thesis*, Stanford University, (1999).
11. T.B. Sebastian, P.N. Klein, B.B. Kimia, "Recognition of shapes by editing shock-graphs," *Eighth International Conference on Computer Vision,*, 1, 755-762, (2001).
12. T. Bernier, J-A. Landry, "A new method for representation and matching shapes of natural objects," *Pattern Recognition*, 36, 1711-1723, (2003).
13. A. Del Bimbo, "Visual Infromation Retrieval," *Morgan Kaufmann, San Franciso, USA*, 56-57, (1999).

Multifocus Image Sequences for Iris Recognition

Byungjun Son[1], Sung-Hyuk Cha[2], and Yillbyung Lee[1]

[1] Department of Computer Science, Yonsei University
134 Shinchon-dong, Seodaemoon-gu, Seoul 120-749, Korea
{sonjun, yblee}@csai.yonsei.ac.kr
[2] Computer Science Department, Pace University,
861 Bedford rd, Pleasantville, NY, 10570, USA
scha@pace.edu

Abstract. We report on an iris recognition system using image sequences instead of single still images for recognition. Image sequences captured at different focus levels provides more information than single still images. Most of the current state-of-the-art iris recognition systems use single still images which are highly focused. These systems does not recognize defocused iris images. The experimental results show that defocused iris images can be correctly recognized if we use multifocus image sequences as gallery images for recognition.

1 Introduction

Automated person authentication systems based on iris recognition are reputed to be the most reliable among all biometric methods because of the high level of stability and distinctiveness of the iris patterns [1]. However, the potential requirement of obtaining high accuracy is that users supply iris images of good quality. Current iris recognition systems require only frontal view images of good quality and remove poor quality images by evaluating qualities of images [2] [3] [4] [5]. The quality factors affecting iris recognition performance are defocus blur, motion blur, off-angle, occlusion, specular reflection, lighting, etc. The defocus blur and motion blur are the quality factors that affect image acquisition time and recognition performance the most. In practice current iris recognition systems continues to grab image frame until it is in focus. This is one of the reasons why iris recognition systems have less user convenience than other biometric systems. To design highly reliable and accurate iris recognition system increasing user convenience, it may be necessary to effectively operate the possibly blurred iris images due to less cooperation of users and camera with low resolution.

In this paper, we present a new scheme for iris recognition based on multifocus image sequences. The multifocus images sequences contain slightly defocused images as well as highly focused image. Unlike the current iris recognition systems, our proposed method can correctly recognize the highly defocused iris images because various information of the iris images captured at different focus levels is used. We also propose a new feature extraction method using direct linear discriminant analysis on wavelet subband to extract discriminative low-dimensional

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 411–420, 2006.

feature vectors. All biometric systems have common disadvantage that the size of the feature set is normally large. It will increase the complexity of computation and make its real-time implementation more difficult and costly. Therefore, it is necessary to extract discriminative low-dimensional feature vectors from the biometric data. Fig. 1 shows the flowchart of the proposed system.

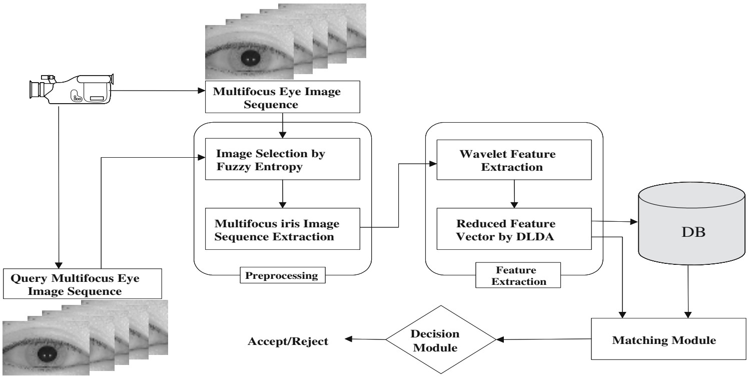

Multifocus Eye Image Sequence
Image Selection by Fuzzy Entropy
Multifocus iris Image Sequence Extraction
Preprocessing
Wavelet Feature Extraction
Reduced Feature Vector by DLDA
Feature Extraction
DB
Query Multifocus Eye Image Sequence
Accept/Reject
Decision Module
Matching Module

Fig. 1. Proposed iris recognition system based on multifocus image sequences

2 Image Acquisition

We first design an image acquisition device to obtain the multifocus image sequences. The CCD camera controlled through RS-485 is used after removing the installed infrared-cutoff filter. The camera is fixed on a pedestal which manually rotates three-dimensionally as shown in Fig. 2. The camera used in our implementation has a 1/4 inch SONY Super HAD CCD with minimum illumination of 0.1 lux and approximately $440,000$ effective pixels minimizing residual image and geometric distortion.

The near infrared light emitting diode(IR-LED) illuminators are in a fixed position to get the same illumination effect over all the images. The use of IR-LED sources with a peak emission wavelength of $780nm$ lead to significant improvement in the quality of the eye image. Therefore, we use two illuminators that each one consists of 36 IR-LED sources with $5.0mm$ diameter and $780nm$ peak wavelength. The imaging of small object like the eye is the source of problems for the usual lenses which are optimized for the distance range of infinity. To capture the adequate eye image, we use a close-up lens called the close-up filter. The close-up lens is attached to the front of the master lens of CCD camera. Any visible light source, such as overhead fluorescent lamps, can produce unwanted corneal reflection lights (glints) and corneal reflection scene on the area of iris. These corneal reflection lights (glints) and corneal reflection scene can degrade performance of the iris recognition system. Therefore, we use the infrared-pass filter (IR-Pass Filter) which filters out visible light below approximately $650nm$ wavelength. It is attached to the front of the close-up lens as shown in Fig. 2.

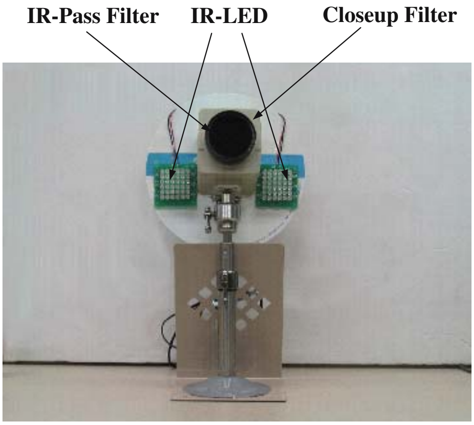


Fig. 2. Image acquisition device capturing the multifocus image sequences

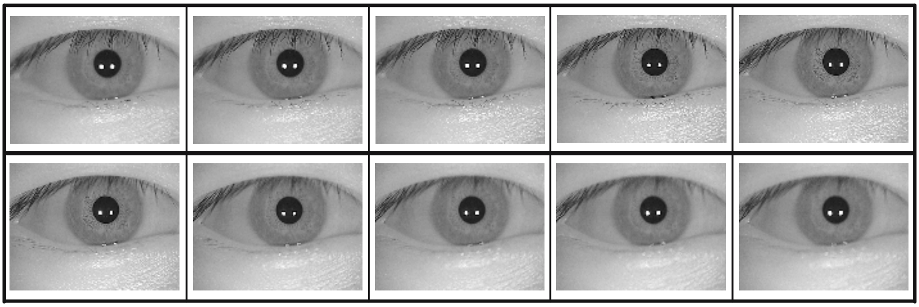


Fig. 3. Example of multifocus image sequence

We obtain the multifocus image sequences by the proposed image acquisition device. The multifocus image sequence is a video sequence with 10 eye images captured at intervals of 0.03 second during changing the focus point from far to near. It contains slightly defocused images as well as highly focused image. Fig. 3 shows sample image sequence containting 10 eye images captured at different focus levels. Each eye image is the 8-bit gray level image with a resolution of 320 × 240.

3 Image Preprocessing

We have obtained a multifocus image sequence with 10 eye images from our proposed image acquisition device. The multifocus image sequence contains various information of eye images captured at different focus levels. However, the iris patterns of some highly defocused eye images may not be localized in preprocessing stage of our proposed system. Processing the multifocus image sequence with the 10 eye images takes too much time to be adapted in real time system like iris recognition system. Therefore, we have selected 5 eye images containing the slightly defocused images as well as a highly focused image with the appropriate quality for the subsequent processing. The image selection will also reduce the process time. We have applied fuzzy set theory for image selection. A notion of entropy in the theory of fuzzy sets was first introduced by de Luca and Termini [6].

For an image F, the fuzzy entropy is defined by the function

$$E(F) = -\sum_{p \in X} (\mu_F(p) log_2 \mu_F(p) + (1-\mu_F(p)) log_2 (1-\mu_F(p))), \tag{1}$$

where $p = (x, y)$ is a pixel in the spacial domain X of the image F and μ_F is a membership function. In this paper, the membership function μ_F is initialized according to

$$\mu_F(p) = \frac{g_p - g_{min}}{g_{max} - g_{min}}, \tag{2}$$

where g_{min} and g_{max} are the minimum and maximum gray levels of the image F respectively and g_p is the gray level of a pixel $p = (x, y)$. Using equation (2) for initializing the membership values has the advantage of stretching out the membership function over the unit interval.

Normalized version $\hat{E}(F)$ of the fuzzy entropy $E(F)$, for which $0 \leq \hat{E}(F) \leq 1$, is clearly given by

$$\hat{E}(F) = \frac{E(F)}{|X|}, \tag{3}$$

where $|X|$ denotes the cardinality of the universal set X, i.e., the size of an image F. Fig. 4 shows fuzzy entropy values of images in sample image sequence. Fig. 5 shows image sequence consisted of 5 images selected from sample image sequence.

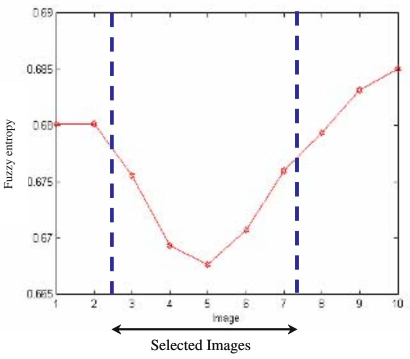


Fig. 4. Fuzzy entropy values of 10 images in sample image sequence

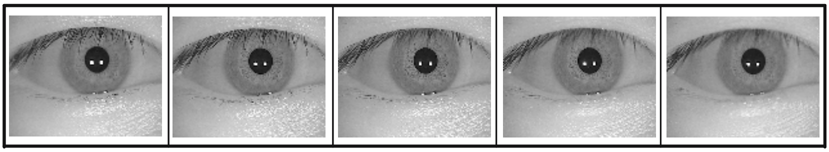


Fig. 5. Images selected from sample image sequence

The most important part of preprocessing is the effective localization of the iris pattern. This localization presents a challenge due to the non-ideality of the available iris images. We use the method for localizing the iris area between the inner boundary and the collarette boundary, to remove unnecessary areas and to increase the recognition rate. Since the pupil area has a low intensity value and looks dark in the eye image, it can be found by edge detector. To find out the inner boundary, Canny edge detector and bisection method are applied to the eye image after excluding unnecessary areas. To exclude unnecessary areas, we first remove light side of the eye image which has gray intensity values of more than mean value. The gray intensity values of between zero and mean value are stretched to value of between 0 and 255. Next, canny edge detector is applied to extract edges of the eye image which is excluded unnecessary areas. A grouping of the connected adjacent edges is accomplished to apply bisection method. The edge components which have the lowest distance coefficient obtained by the bisection method indicate an inner boundary. To find out an outer boundary, canny edge detector is applied to the original eye image with different parameters again. The shortest distance between the center of the pupil and the edges of the lower eyelid is set to the radius of outer boundary.

The collarette is the dividing line between the pupillary zone and ciliary zone of the iris. The pupillary zone is vastly various and the ciliary zone is seldom various. To locate the collarette boundary, the iris area between the inner boundary and the outer boundary is converted cartesian coordinate system into polar coordinate system. Histogram equalization is applied to the converted iris image and the pixels which have seldom variety are removed by highpass filter with one-dimensional Fourier transform. Finally, collarette boundary is found by using

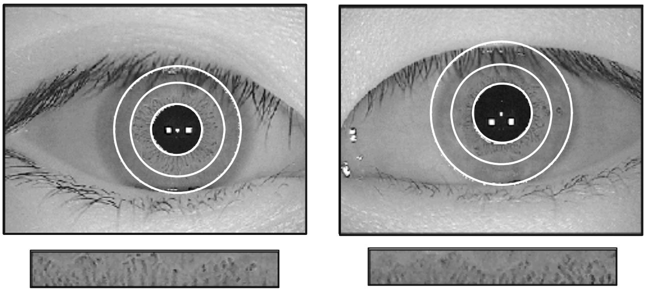


Fig. 6. Examples of the detected boundaries and the localized iris area

statistical information. We make a 225×3 window from the image and calculate mean values in each window. The magnitude of difference at adjacent pixels is represented by the mean value. The rate of cumulative mean value is also taken from the mean value. From the rate of cumulative mean value, standard deviation value is also derived. The first position where the standard deviation value converges to the zero is a collarette boundary. Fig. 6 shows the inner boundary, outer boundary and collarette boundary detected in the eye image and the localized iris area.

4 Feature Extraction and Pattern Matching

Traditionally, to represent the biometric trait as low dimensional feature set, principal component analysis (PCA) and linear discriminant analysis (LDA) are performed on the biometric image or signal. However, these methods have their limitations: poor discriminatory power and large computational load. In view of these limitations, we propose a new approach in using direct linear discriminant analysis (DLDA) - apply DLDA on wavelet subband.

Wavelet transform (WT) is an increasingly popular tool in image processing and computer vision in the past ten years. Wavelet transform has the nice features of space-frequency localization and multiresolutions [7] [8] [9]. The main reasons for popularity of wavelet transform lie in its complete theoretical framework, the great flexibility for choosing bases and the low computational complexity. In our proposed method, The three-level lowest frequency subband with resolution of 29×4 is selected to extract the wavelet feature vector. Generally, low frequency components represent the basic figure of an image, which is less sensitive to varying images. These components are the most informative subimages gearing with the highest discriminating power. After extraction of the wavelet feature vector, the original iris vector x of $7,200$ dimensions is transformed to the feature vector y of 116 dimensions. To reduce the feature dimensionality further and enhance the class discrimination, we apply a direct linear discriminant analysis algorithm (DLDA) proposed by Yu and Yang. The key idea of their method is to discard the null apace of S_b by diagonalizing S_b first and then diagonalizing S_w. As pointed out by Yu and Yang the traditional LDA procedure takes the revers order and consequently discards the null space of S_w which contains discriminative information. This diagonalization process also avoids the singularity problems related to the use of the pure LDA in high dimensional data where the within-class scatter matrix S_w is likely to be singular [10][11].

To apply the proposed method to the iris recognition system, we need two stages, namely, training and recognition stages. In the training stage, we first need a training set composed of a relatively large group of subjects with iris characteristics. The appropriate selection of the training set directly determines the validity of the final results. Second, wavelet transform (WT), using the biorthogonal spline wavelet of order (1, 3), is applied to decompose the iris images for each subject in the training set. We construct the lexicographic vector expansion $\phi \in R^{29 \times 4}$. This vector corresponds to the wavelet feature vector representation of the iris. In the third step, DLDA is applied on the wavelet feature vector with dimension 116 to find the transformation matrix P_{dlda}. Next, the wavelet feature vectors (WFVs) are transformed into the reduced wavelet feature vectors (RWFVs) by the transformation matrix P_{dlda}. Finally, the reduced wavelet feature vectors (RWFVs) and the transformation matrix P_{dlda} are stored in the database. When an unknown iris image x is presented to the recognition stage, wavelet transform is applied to obtain the wavelet feature vector in the same way as the training stage. The wavelet feature vector is transformed into a reduced wavelet feature vector by the transformation matrix P_{dlda} in the database. Finally similarity measurement between the probe RWFV and the reference RWFVs in the database is taken to determine which one of reference images in the database best matches the input probe image. Euclidean distance is the most common and simple similarity measurement. We have used Euclidean distance because it suit well with DWT+DLDA method and is very fast and simple. The matching distances of the probe RWFVs extracted from multifocus image sequence are computed with all the reference RWFVs and the minimum distance is taken as the final value.

5 Experimental Results

The experiments in this section are performed on the YSUIDB2005 database. The YSUIDB2005 database contains $13,750$ eye images of 110 classes, with 25 multifocus image sequences per class. Each multifocus image sequence contains 5 eye images captured at different focus levels. The eye images were captured from students of the Yonsei University by our proposed image acquisition device. The size of an image captured by an image acquisition device is 320×240 pixels. The YSUIDB2005 database consists of the multifocus image sequences containing slightly defocused eye images. we randomly choose 15 multifocus image sequences per class (person) for training, the other 10 multifocus image sequences for testing. To show the statistical robustness of proposed methods, we employed the cross validation sampling technique where all recognition rates were determined by averaging 20 different rounds of iris recognition.

Identification performance can also be quantified by method based on top N matches. Here, the probability of the true users template (sample) being selected at least once in the top N choices (that the system gives as output) is considered. For example, if there is a huge database of users to be compared against in an identification scenario, then the system comes up with a smaller list of N possible matches (N candidates). Even if one of the templates from the possible matches corresponds to the true user, we say that identification is achieved. This kind of matching is more useful and reasonable where there is a human inspector to make a final decision. The purpose is to reduce the workload of the human inspector.

Table 5 shows identification rates for N candidates in feature space with d dimentions, where N is the number of top matches selected. The best identification rates for 1 candidate 2 candidates are 99.83% and 99.92% for dimension

Table 1. Changes of recognition rates according to the number of candidate and feature dimension

Feature Dimension	Recognition Rate(%)			
	1 Candidate	2 Candidate	5 Candidate	10 Candidate
5	85.06	93.92	98.63	99.46
10	98.74	99.18	99.70	99.89
15	99.37	99.52	99.84	99.94
20	99.65	99.70	99.84	99.93
25	99.70	99.83	99.91	99.94
30	99.72	99.85	99.89	99.96
35	99.70	99.81	99.89	99.90
40	99.78	99.82	99.87	99.95
45	99.77	99.86	99.95	99.96
50	99.81	99.85	99.95	99.97
55	99.83	99.90	99.93	99.96
60	99.80	99.83	99.94	99.95
65	99.75	99.79	99.93	99.96
70	99.80	99.87	99.95	99.98
75	99.77	99.88	99.92	99.99
80	99.77	99.92	99.94	99.98

$d = 55$ and dimension $d = 55$, respectivley. For 5 candidates, the best identification rate is 99.95% for dimension $d = 45$. It is 99.99% for candidate $N = 10$ and dimension $d = 75$.

Verification is process of comparing a submitted biometric sample against biometric reference of a single enrollee whose identity or role is being claimed. In principle, two commonly used error measures for a verification system are False Acceptance Rate(FAR) - an imposter is accepted - and False Rejection Rate(FRR) - an authentic individual is rejected. Fig. 7 show authentic distributions and impostor distributions of verification systems for YSUIDB2005 database. A false rejection rate (FRR) and a false acceptance rate (FAR) are 0.7% and 0.1% for threshold 0.13.

To show that the highly defocused images may be recognized correctly, we have captured 10 highly defocused images for each class. The YSUIDB2005 database does not contain the highly defocused images. Table 5 shows comparison of recognition rates using the highly defocused images as test data. The tests were conducted in each case using multifocus image sequence and only the n^{th} image in each multifocus image sequence as training data. 1^{st} image and 5^{th} indicate the most focused image and the most defocused image of five images contained the multifocus image sequence, respectively. In the case of matching each highly defocused single image against each multifocus image sequence representing each person, the average recognition rate is 99.79% with feature vector consisted of 55 components. The average recognition rates are 97.62% and 98.99% with same dimension $d = 55$ when we match each highly defocused single image against each 1^{st} image and 5^{th} image, respectively. The recognition rate for highly defocused images goes down when the quality of gallery images goes up. In these experiments, we find that the highly defocused image may be correctly recognized if we have multifocus image sequences as gallery (reference) images.

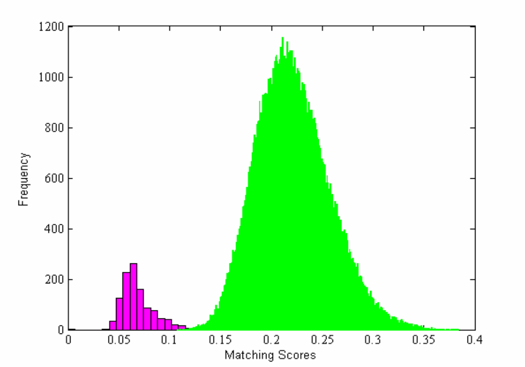
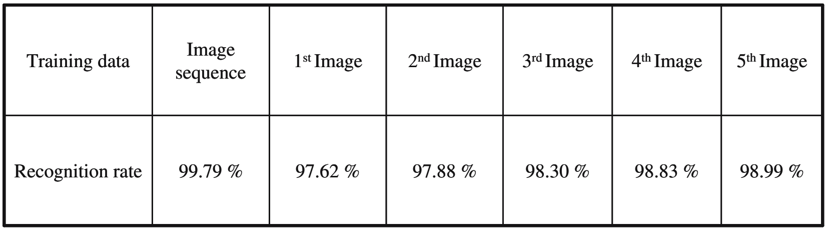


Fig. 7. Distribution of matching distances

Table 2. Comparison of recognition rates using highly defocused images as test data. The tests were conducted in each case using only 1^{st} image, only 2^{nd} image, only 3^{rd} image, only 4^{th} image, only 5^{th} image, and multifocus Image sequence as training data.



Training data	Image sequence	1st Image	2nd Image	3rd Image	4th Image	5th Image
Recognition rate	99.79 %	97.62 %	97.88 %	98.30 %	98.83 %	98.99 %

6 Conclusion

In this paper, a new framework for iris recognition using multifocus image sequence has been proposed upon which future recognition systems can be built. In addition, we designed an image acquisition device to capture the multifocus image sequences. The key feature of this new iris acquisition device is that it integrates and simply aligns multiple optical elements greatly reducing both the material and labor costs of the iris recognition system. A new feature extraction method based on wavelet transform and direct linear discriminant analysis (DLDA) has been proposed in this research. By these proposed methods, the best identification rate was 99.83% when the number of features is 55. For a verification system , a false rejection rate (FRR) and a false acceptance rate (FAR) were 0.7% and 0.1% for threshold 0.13, respectively. We have also confirmed that the iris recognition system using the multifocus image sequence may work well on defocused iris images as well as highly focused iris image.

Acknowledgements

This research was supported as a Brain neuroinformatics Research program by Korean Ministry of Commerce, Industry, and Energy.

References

1. J. G. Daugman: “Statistical richness of visual phase information,” *Internaltional Journal of Computer Vision*, vol.45, no. 1, pp. 25-38, 2001.
2. J. G. Daugman: “High confidence visual recognition of persons by a test of statistical indenpendence,” IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 15, pp.1148-1161, 1993.
3. R. P. Wildes: “Iris Recognition : An emerging biometric technology,” Proc. of the IEEE, 85(9):1348-1363, 1997.
4. M. Negin, T. Chmielewski, M. Salgnicoff, U. von Seelen, P. Venetainer, and G. Zhang: “An iris biometric system for public and personal use,” In IEEE Computer, vol. 33, pp. 70-75, 2000.
5. L. Ma, T. Tan, Y. Wang, and D. Zhang: “Personal identification based on iris texture analysis,” IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 15, pp.1148-1161, 1993.
6. A. De Luca and S. Termini: “Definition of a non probabilistic entropy in the setting of fuzzy sets theory,” IEEE Computer, vol.24, pp. 55-68, 1972.
7. J. H. Lai, P. C. Yuen, and G. C. Feng: “Face recognition using holistic Fourier invariant features,” Pattern Recognition, vol.34, pp. 95-109, 2001.
8. C. Garcia, G. Zikos, and G. Tziritas: “Wavelet packet analysis for face recognition,” Image Vision Computing, vol.18, pp. 289-297, 2000.
9. I. Daubechies,: “The wavelet transform, time-frequency localization and signal analysis,” IEEE Trans. Information Theory, vol.36, no. 5, pp. 961-1005, 1990.
10. H. Yu and J. Yang, : “A Direct LDA Algorithm for High-Dimensional Data - with Application to Face Recognition,” Pattern Recognition, 34(10):2067–2070, 2001.
11. K. Fukunaga: “Introduction to Statistical Pattern Recognition,” Academic Press, 1990.

Projection Method for Geometric Modeling of High Resolution Satellite Images Applying Different Approximations

Fahim Arif[1], Muhammad Akbar[2], and An-Ming Wu[3]

[1] Computer Science Department, College of Signals
National University of Sciences and Technology, Rawalpindi, Pakistan
fahim-mcs@nust.edu.pk
[2] Chief Instructor (Engineering Division), College of Signals
National University of Sciences and Technology, Rawalpindi, Pakistan
makbar-mcs@nust.edu.pk
[3] National Space Organization, Hsinchu, Taiwan
amwu@nspo.org.tw

Abstract. Precise remote sensing and high resolution satellite images have made it necessary to revise the geometric correction techniques used for ortho-rectification. There have been improvements in algorithms from simple 2D polynomial models to rigorous mathematical models derived from digital photogrammetry. In such scenario, conventional methods of photogrametric modeling of remotely sensed images would be insufficient for mapping purposes and might need to be substituted with a more rigorous approach to get a true orthophoto. To correct geometric distortions in these, the process of geometric modeling becomes important.

Pixel projection method has been devised and used for geometric correction. Algorithm has been developed in C++ and used for FORMOSAT-2 high resolution satellite images. It geo-references a satellite image while geo-locating vertices of the image with its geo-locations extracted from ancillary data. Accuracy and validity of the algorithm has already been tested on different types of satellite images. It takes a level-1A image and the output image is level-2 image. To increase the geometric accuracy, a set of ground control points with maximum accuracy can also be selected to determine the better knowledge of position, attitude and pixel alignment.

In this paper, we have adopted different techniques of approximations and applying three possible methods of interpolation for transformations of image pixels to earth coordinate system. Results show that cubic convolution based modeling gives best suitable output pixel values while applying transformation.

Keywords: Geometric modeling; Geo-referencing; Geometric correction; Pixel projection; Remote Sensing; High resolution; 2-D Interpolation; Cubic convolution.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 421 – 432, 2006.

1 Introduction

Images taken by high resolution optical satellite sensors are available with different product levels. They are starting with the original images over close to original images, improved by the sensor calibration, projections to a geodetic plane and orthoimages based on digital elevation models (DEMs) [1, 2]. Remote sensing satellites with very high resolution optical systems also have precise attitude determination systems. These systems allow precise geo-referencing. Images projected to a plane with constant height or a DEM do require a different mathematical model for accurate geo-referencing [3-8]. For some such images no information about the sensor geometry is available, so a geometric reconstruction and a mathematical model have to be devised. The geometric models for handling of these images are well known, but some have to be improved by adapting additional parameters for an optimal solution [9,10].

Remotely sensed images are available as different geometric products, ranging from original sensor images, just improved by inner platform orientation named as level-1A products, over projections to a world coordinate system named as level-2 products and up to ortho-rectified images [5, 7, 11]. Different mathematical models are used for their handling [12-15]. Similarly different solutions are being used like an improvement of the pre-orientation say just a shift in X and Y coordinates not taking advantages from the given sensor orientation [16] to feature [17] and contour matching [6, 9] using high image processing methods etc.

Pixel projection method has been devised by National Space Program Office (NSPO) Taiwan, for processing of level-1 satellite images. This algorithm takes into account four vertices of the satellite image and geo-reference them to ground coordinates. It uses WGS84 as geodetic model and extracts details of vertices from ancillary data of the image, available in DIMAP format. This model is used for processing of FORMOSAT2 images [13, 14].

In this paper transformation of image pixels have been transformed using different 2-D interpolation methods. Conventionally each approximation method has its own merits and demerits but they are tested separately, so that to arrive at a best suitable result. In this way algorithm for level-3 geometric correction can be applied more concretely [18]. Organization of paper is: section 2 describes factors affecting satellite image geometry, section 3 gives details of method of pixel projection applied for geometric modeling of satellite images, section 4 gives the ancillary data and images used for testing purposes. Section 5 describes outcome of applying different approximations techniques and in section 6 results of all three methods are discussed.

2 Factors Affecting Satellite Image Geometry

2.1 Parameters of Orbit Model

The orbits of most earth remote sensing satellites are nearly circular because a constant image scale is desired. The orbital velocity of satellites can be considered constant in time e.g 1.0153 x 10^{-3} radians/second for Landsat-1/ 2. Variations in platform altitude and ground speed cannot be assumed negligible [20]. Platform attitude is critical to geometric precision because of the long "moment arm" of high altitude satellite pointing. A very small change in the pointing angle results in a large change in the viewed location on the ground. Attitude is expressed by three angles of platform rotation: roll, pitch and yaw, shown in fig 1.

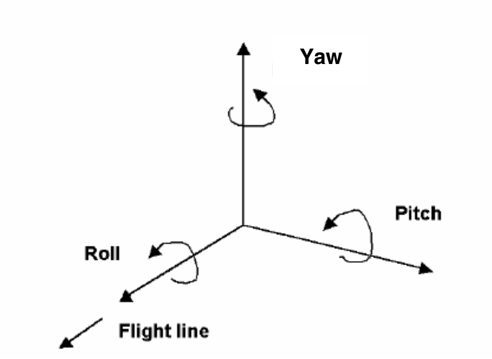


Fig. 1. Platform attitude angle definition

2.2 Earth Model

Although the earth's geometric properties are independent of the sensor's, they interact intimately via the orbital motion of the satellite. There are two factors to consider, one that earth is not exact sphere but oblate described by equation 1 and second that the earth rotates at a constant angular velocity ωₑ (given in equation2) while the satellite is moving along its orbit and scanning orthogonal.

$$\frac{p_x^2 + p_y^2}{r_{eq}^2} + \frac{p_z^2}{r_p^2} = 1 \qquad (1)$$

Where (p_x,p_y,p_z) are the geocentric coordinates of any point P on the surface of the earth, r_{eq} is the equatorial radius and r_p is the polar radius [3].

$$v_0 = \omega_e r_e \cos \varphi \qquad (2)$$

Where r_e is the earth's radius and φ is the geodetic latitude [3].

3 Method of Pixel Projection for Geometric Modeling

Geometric model of a satellite image is processed for finding the position of each pixel in geographical or map coordinate system. The direct geo-referencing method gains its advantage with increasing satellite ancillary data accuracy [13,14]. For example, the FORMOSAT-2 can provide position data with accuracy up to 20 meters. The method is as follows:

The first step is: given satellite state vectors and attitude quaternion at a certain instant of time, the intersection of the sensor line-of-sight with the reference earth ellipsoid can be found by applying a simple algorithm involving several coordinate transformations and basic geometry. A schematic is shown in fig. 2. DEM is not used in this step.

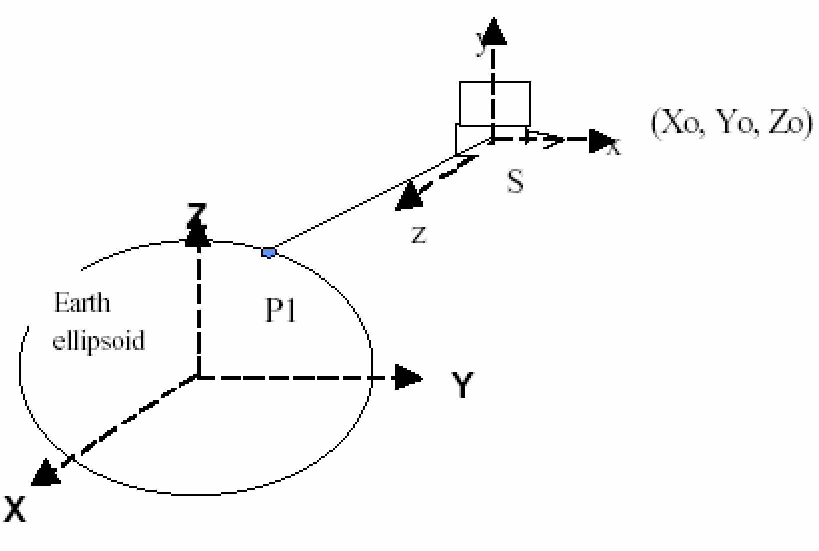


Fig. 2. Schematic for georeferencing without DEM

The satellite ancillary data gives time of imaging in UTC, spacecraft orbital position and velocity in ECF, and the ECI to body quaternion. The procedure for pixel projection in ECF is as follows:

Step 1: compute Earth rotation matrix (J2000 to ECF) based on the Julian day number corresponding to the UTC time.
Step 2: transform the attitude (in terms of Euler angles in the sequence of 1-2-3) from body to LVLH, to ECI, and to ECF and compute the corresponding “body to ECF” transformation matrix.
Step 3: transform sensor line-of-sight vector to ECF.
Step 4: find the coordinate (latitude, longitude, height) of the intersection of sensor line-of-sight vector and the earth ellipsoid surface (WGS84).

The pixel position in ECF coordinate (*X,Y,Z*) can be found by solving the following equations:

$$\frac{X^2}{a^2}+\frac{Y^2}{a^2}+\frac{Z^2}{b^2}=1 \tag{3}$$

$$\frac{X-X_0}{p}=\frac{Y-Y_0}{q}=\frac{Z-Z_0}{r} \tag{4}$$

Where *a* is the earth ellipsoid’s semi-major axis, *b* is the semi-minor axis, (*p,q,r*) is the elements of the CCD line-of-sight vector in ECF coordinate. (*Xo,Yo,Zo*) is the spacecraft position in ECF coordinate.

4 Ancillary and Test Data

Test images 1 and 2 shown in fig. 3 were obtained from FORMOSAT-2 (courtesy NSPO). Test image 1 represents general area south of Islamabad, image was taken on 28th Oct 2005 and test image 2 represents remote area of Abbotabad, image was taken on 9th Oct 2005 one day later than severe earthquake struck the northern parts of Pakistan. Important features extracted from the ancillary data (available in dim file format) are given in table 1.

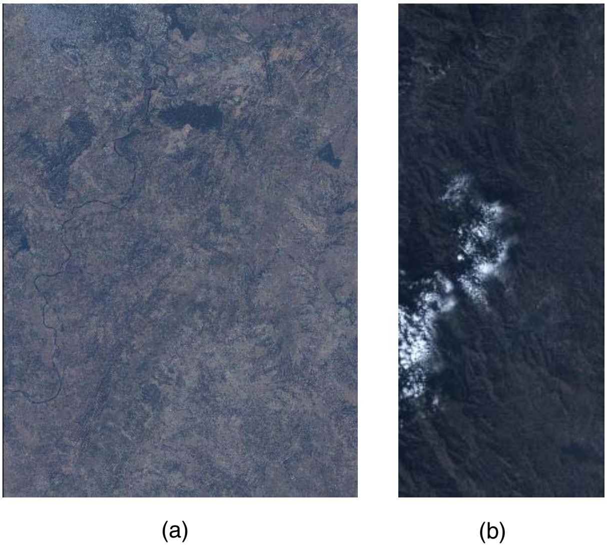


Fig. 3. FORMOSAT-2 Images obtained from NSPO for testing purpose (a) Test image1 showing area south of Islamabad (b) Test image2 showing remote area of Abbotabad

Table 1. Ancillary Data of the Test Images

META DATA	Test Image 1	Test Image 2
METADATA_FORMAT	DIMAP	DIMAP
METADATA_PROFILE	F2_Level_1A	F2_Level_1A
DATASET_NAME	FS2_112627000_1A_0001_MS	FS2_114586000_01_0001_MS
COPYRIGHT	COPYRIGHT 2004/09	COPYRIGHT 2004/09
SCENE_ORIENTATION	194.22	194.18
DATASET_PRODUCER_NAME	NSPO	NSPO
DATASET_PRODUCER_URL	http://www.nspo.org.tw	http://www.nspo.org.tw
IMAGING DATE	2005/10/28	2005/10/09
PRODUCT_INFO	FORMOSAT2 Product Level1A MS	FORMOSAT2 Product Level1A MS
RASTER DIMENSIONS		
No of Columns	3000	1500
No of Rows	4200	3600
PROCESSING_LEVEL	1A	1A
HORIZONTAL_CS_NAME	WGS 84	WGS 84

Two test images are processed and geo-referenced using pixel projection method. The output images are level 2 images and are shown in fig. 4. Pixel projection method takes into account WGS84 world geodetic system for geo-referencing satellite data to an earth coordinate system. Original images are primarily ortho-corrected selecting vertices from the ancillary data.

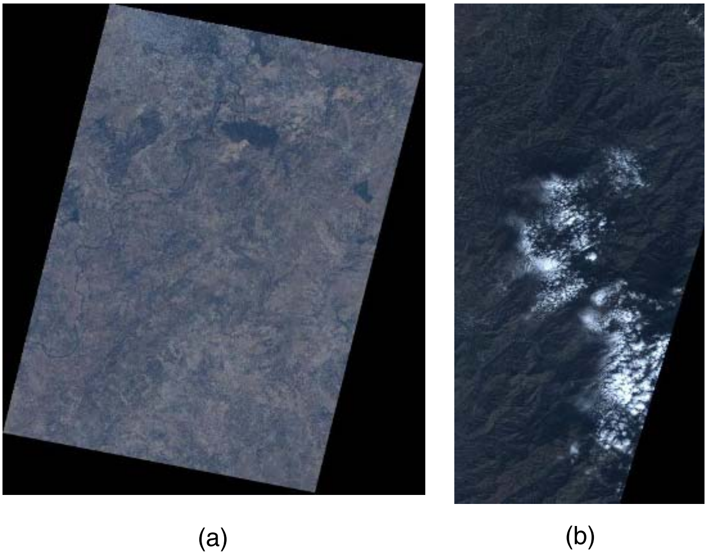


Fig. 4. Geometrically corrected images using pixel projection method (a) Image of south of Islamabad (b) Abbotabad image

5 Applying Different Approximations for Pixel Transformations

Each approximation scheme has distinct properties and effects on an image's information content, but each method must be evaluated with respect to the imagery's intended use [19]. The methods are:

5.1 Nearest Neighbor

Each output pixel value in this method is the unmodified value from the closest input pixel. Less computation is involved than in other methods, leading to a speed advantage. However, the raster data if is resampled to a different cell size, a blocky appearance can result from the duplication (smaller output cell size) or dropping (large cell size) of input cell values.

5.2 Bilinear Interpolation

An output cell value in the bilinear interpolation method is the weighted average of the four closest input cell values. This method produces a smoother appearance than the nearest neighbor approach, but it can diminish the contrast and sharpness of feature edges.

5.3 Cubic Convolution

This method calculates an output cell value for a 4 x 4 block of surrounding input cells. This method produces sharper, less blurry images than bilinear interpolation, but it is the most computationally intensive resampling method [21].

5.4 Comparison

Figs. 5 and 7 are the illustration of results applying above mentioned three schemes of approximations. Nearest neighbor, bilinear and cubic convolution were applied one by one, and these results were gained. In both the figures (a), (b) and (c) show the transformation of the pixel projection algorithm obtained using these approximations respectively. Whereas figure 6 and 8 are 3-dimensional plots of these interpolating approximations representing nearest neighbor, bilinear and cubic convolution in (b) (c) and (d) respectively. They represent comparison of 2-D interpolation methods on a 7 by 7 matrix data. Figure 6 (a) and 8 (a) are 3-D plot meshes obtained applying MATLAB meshgrid function for ranges of (-3: 0.5: 3) and (-3: 1: 3) respectively which transforms the domain specified by vectors x and y to arrays X and Y. To generate peaks at low resolution MATLAB's *peaks* function was used, $z = peaks(x, y)$.

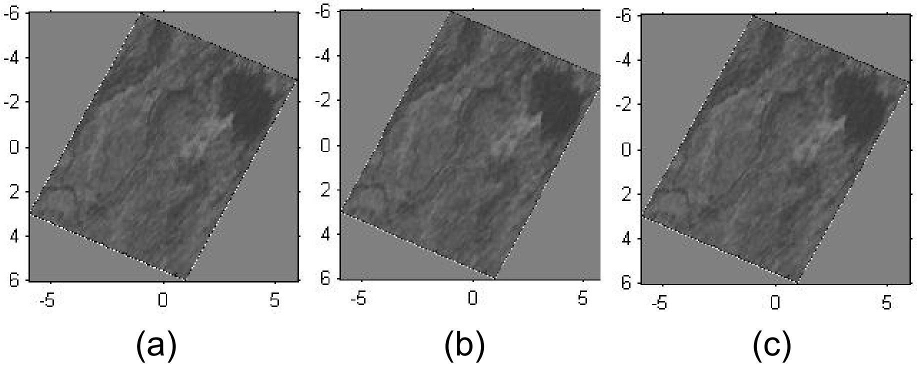


Fig. 5. Results of three different approximation techniques on South of Islamabad image (a) Using nearest neighbor method (b) Using bilinear interpolation method (c) Using cubic convolution method

Comparison of surface plots of different interpolation methods reveal that the bicubic method produces smoother peaks than nearest and bilinear interpolation methods. The information loss is negligible in this method and remains closer to the original image data. Surface plots for these methods were obtained after applying MATLAB function *imtransform* which transforms the input image A according to the

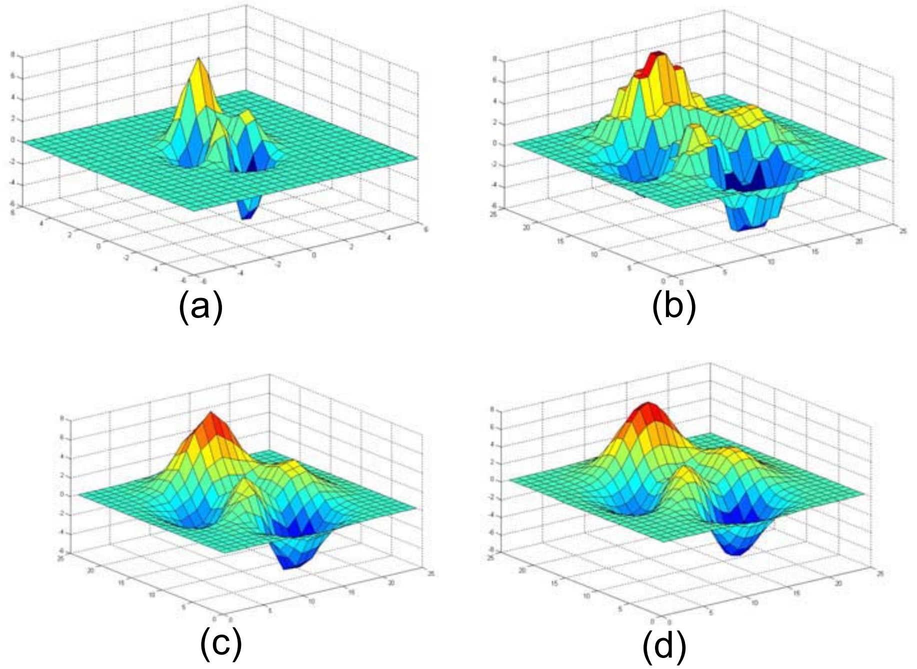


Fig. 6. 3-Dimensional plots of the approximations applied on Test image 1 South of Islamabad image (a) Meshgrid (b) Nearest neighbor (c) Bilinear interpolation (d) Cubic convolution

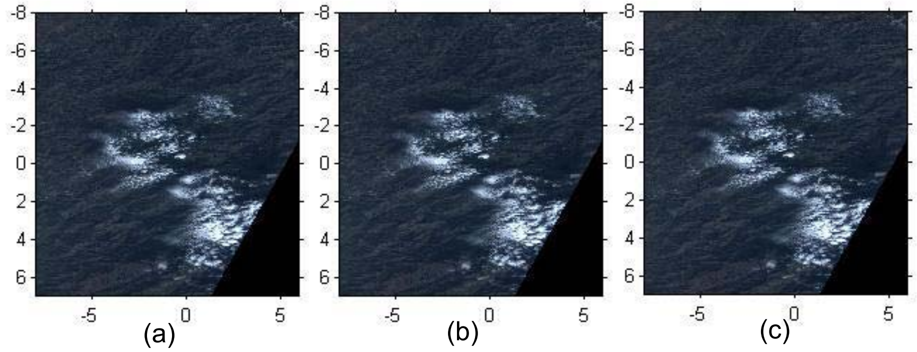


Fig. 7. Results of three different approximation techniques on Abbotabad image (a) Using nearest neighbor method (b) Using bilinear interpolation method (c) Using cubic convolution method

2-D spatial transformation defined by *TFORM*, which is a spatial transformation structure (TFORM) as returned by *maketform* or *cp2tform*. In this function following specific parameters were used to control various aspects of the spatial transformation. These parameters are listed in table 2.

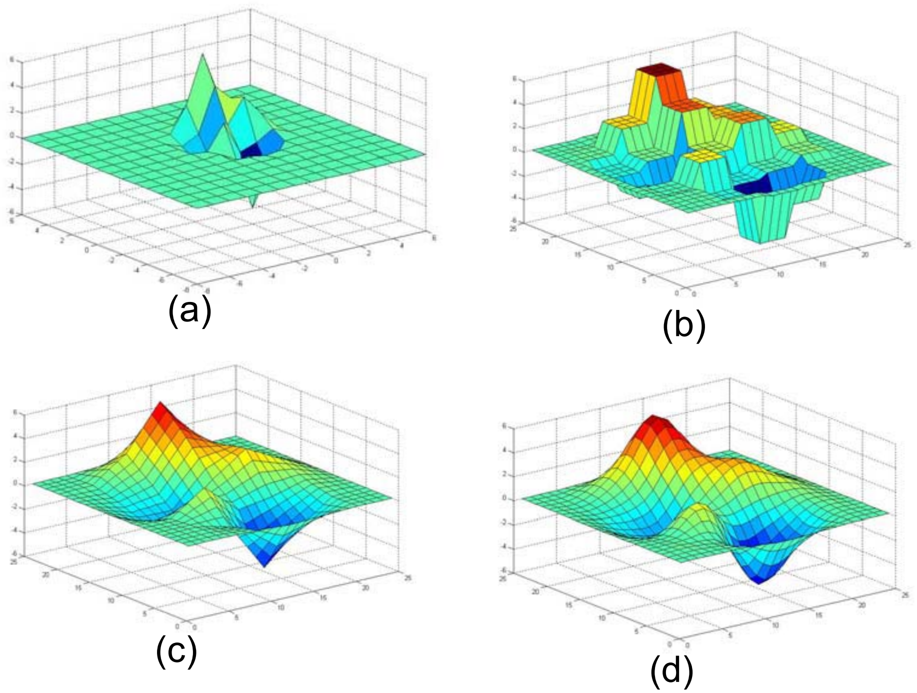


Fig. 8. 3-Dimensional plots of the approximations applied on test image 2, image of Abbotabad (a) Meshgrid (b) Nearest neighbor (c) Bilinear interpolation (d) Cubic convolution

Table 2. Descriptions of Parameters used in Programming

Parameter	Description
UData VData	These are two real vectors specifying spatial location of input image in the 2-D space U-V. The two elements of 'UData' give the u-coordinates and 'VData' give the v-coordinates, respectively. The values used here are [1, 1].
XData YData	These are two real vectors specifying spatial location of output image in the 2-D space X-Y. The two elements of 'XData' give the x-coordinates and 'YData' give the y-coordinates, respectively. The values used here are [-6, 6] for Islamabad image and [-8, 7] for Abbotabad iamge.
XYScale	Specifies the width of each output pixel in X-Y space.
Size	Specifies the number of rows and columns of the output image. Here size is the same as input image.
FillValues	For gray scale images specified as 128.

6 Results and Discussion

Pixel projection method devised for geometric correction of high resolution satellite images produces optimized results in its application. A raw satellite image defined as level-1A images is processed significantly to produce a level-2 image. Algorithm has been further developed and processing of level-3 image is achieved [18].

Transformation has been done with bilinear interpolation previously. There was a dire need of investigation that which 2-D interpolation is more suitable for transformation of pixel values. Therefore in this paper we applied three possible methods of approximations to obtain more optimal results. Two test images of FORMOSAT-2 satellite were used taken in recent past.

Present application is developed in MATLAB while adopting certain inbuilt set of functions and tools. They include *cpselect, cpstruct2pairs, cp2tform, tformfwd,* and *imtransform* from the image processing toolbox, in combination with pix2map, *meshgrid* and interpolation functions like *interp2* from the mapping toolbox together with mathworks.com's help (www.mathworks.com) which enabled us to geo-register remotely sensed data based on control point pairs to carry out testing and comparison of different interpolation techniques.

With the above mentioned methodology, three different approximation techniques were applied on two different test images shown in figure 3. The resultant images of nearest neighbor, bilinear interpolation and cubic convolution approximations were displayed in figs 5 and 7. Their 3-D plots (figs. 6 and 8) were obtained to see the surfaces of these functions showing smoothness in peaks. The 3-D plot of cubic convolution showed the most suitable peaks curve and hence is considered the excellent choice for pixel value transformation. To prove the same result alternatively,

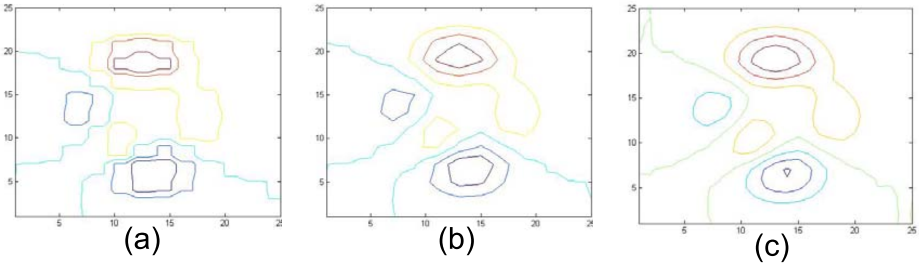


Fig. 9. 2-D contour plots for approximations techniques applied on Islamabad image, showing (a) Nearest neighbor method (b) Bilinear interpolation method (c) cubic convolution method

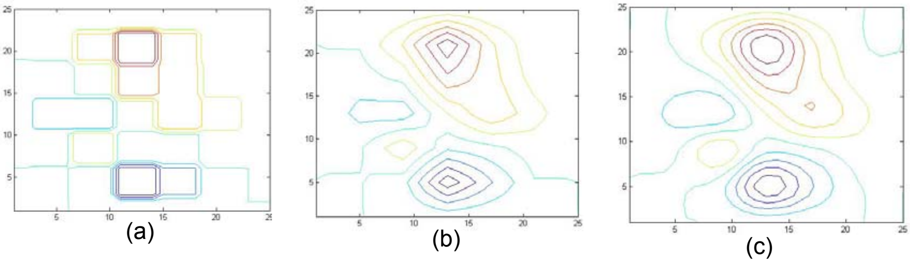


Fig. 10. 2-D contour plots for approximations techniques applied on Abbotabad image, showing (a) Nearest neighbor method (b) Bilinear interpolation method (c) cubic convolution method

another good function of MATLAB is used. *imcontour* draws a 2-D contour plot of the intensity image, automatically setting up the axes so that their orientation and aspect ratio match the image. Contour plots for all three interpolation techniques are shown in figs 9 and 10. Comparison of contour plots based on the surface plots of these approximation methods show the minimum and maximum values of the input image data matrix. It is evident that contour plots for cubic convolution method in figs 9 (c) and 10 (c) give more smoother details of image data.

7 Summary

Different mathematical and geometrical models are used for the handling of satellite image geometry. Pixel projection method was developed for geometric correction of high resolution satellite images. A level-1A image is processed significantly to produce a level-2 image. This algorithm takes into account four vertices of the satellite image and geo-reference them to ground coordinates system based on WGS84 geodetic model and details from ancillary data. This model is used for processing of FORMOSAT-2 images.

In this paper transformation of image pixels have been transformed using different 2-D interpolation methods. Each approximation method with its merits and demerits has been discussed and tested. Previously, transformation was done with bilinear interpolation. So there was a need for investigation of other 2-D interpolation method as well for pixel value transformation. Therefore we applied three possible methods of approximations to obtain more optimal results. Two test images of FORMOSAT-2 satellite were taken as input images.

Three different approximation techniques (nearest neighbor, bilinear interpolation and cubic convolution) were applied. The resultant images of these methods and their surface and contour plots were shown in figs. 5 to 10. The 3-D plot of cubic convolution method showed the most suitable peaks curve. Same result was proved with contour analysis of the functions also. Cubic convolution is the best approximation technique available for pixel projection method to be adopted for pixels value transformation.

References

1. Dial. G, and J. Grodecki, Satellite Image Block Adjustment Simulations with Physical and RPC Camera Models, ASPRS Annual Conference Proceedings, Colorado, (2004).
2. W. Zhang and Z. Li, “Rectification of airborne line-scanner imagery utilizing flight parameters”, First International Airborne Remote sensing conference and exhibition, Strasbourg, France, (1994).
3. R. A. Schowengerdt, Remote sensing models and methods for image processing, Second Edition, Academic Press, Elsevier, USA, (1997).
4. T. M. Lillesand and R. W. Kiefer, Remote sensing and image interpretation, Fourth Edition, John Wiley and sons inc, Singapore, (2003).
5. P. R. Wolf and Bon A. dewitt, Elements of photogrammetry with applications in GIS, Third Edition, Mcgraw Hill, USA, (2000).

6. F. Eugenio, Automatic satellite image georeferencing using a contour-matching approach, IEEE transactions on geoscience and remote sensing, vol. 41, No. 12, (2003).
7. X. Dai and S. Korran, A feature-based image registration algorithm using improved chain-code representation combined with invariant moments, IEEE Trans. Geosci. Remote Sensing, vol. 37, (1999), 2351–2362.
8. Z. Mao, D. Pan, H. Huang, and W. Huang, Automatic registration of SeaWiFS and AVHRR imagery, Int. J. Remote Sens., vol. 22, (2001), 1725–1735.
9. F. Eugenio, F. Marqués, and J. Marcello, Pixel and sub-pixel accuracy in satellite image georeferencing using an automatic contour matching approach, in Proc. IEEE Int. Conf. Image Processing, (2001), I.822–I.825.
10. F. Eugenio, J. Marcello, F. Marqués, A. Hernández-Guerra, and E. Rovaris, A real-time automatic acquisition, processing and distribution system for AVHRR and SeaWiFS imagery, IEEE Geosci. Remote Sensing Newslett., no. 120, (2001) 10–15.
11. M. N. Demers, Fundamental of Geographic Information Systems, Second Edition, John Wiley and sons inc, Singapore, (2002).
12. H. S. Stone and R. Wolpov, Blind cross-spectral image registration using prefiltering and fourier-based translation detection, IEEE Trans. Geosciences and Remote Sensing, vol. 40, (2002) 637–650.
13. A. M. Wu and Y. Y. Lee, Geometric correction of high resolution image using ground control points, proceedings of 22nd Asian conference on remote sensing, Singapore, (2001).
14. Y. Y. Lee, A. M. Wu and F. Wu, An algorithm for geometric correction of high resolution image based on physical modeling, website, www.gisdevelopment.net/aars/acrs/2002.
15. J. Le Moigne,W. J. Campbell, and R. F. Cromp, An automated parallel image registration technique based on the correlation of wavelet features, IEEE Trans. Geosci. Remote Sensing, vol. 40, (2002) 1849–1864.
16. J. R. Schott, Remote sensing-The image chain approach, Oxford University Press, new York, (1997).
17. M. Erdogan, O. Eker, A. Yilmaz and O. Aksu, Orthorectification of SPOT images with the same-pass constraints, XXth ISPRS congress, Istanbul, Turkey, (2004).
18. F. Arif, M. Akbar and A. M. Wu, Most Favorable Automatic Georeferencing Based on GCPs Selection using Least Square Method, in Proceedings of The 4th International Conference on Digital Earth, Taipei, Taiwan, (2006), 183-192.
19. B. R. Corner, R. M. Narayanan and S. E. Reichenbach, Noise Estimation in Remote Sensing Imagery using Data Masking, International Journal of Remote Sensing,Vol. 24, No. 4, (2003) 689–702.
20. Di, K., R. Ma and R. Li, Rational functions and potential for rigorous sensor model recovery. Photogrammetric Engineering and Remote Sensing, 69(1), (2003), 33-41.
21. F. Arif and M. Akbar, A New Approach for Anti-aliasing Raster Data in Air Borne Imagery, in Proceedings of SPIE Vol. 5969, Photonic North International Symposium, Toronto, Canada, (2005), 59692L-1 to9.

Global Localization of Mobile Robot Using Omni-directional Image Correlation

Sukjune Yoon, Woosup Han, Seung Ki Min, and Kyung Shik Roh

Mechatronics & Manufacturing Technology Center Samsung Electronics CO., LTD.416
Maetan-3Dong, Yeongtong-Gu, Suwon Shi, Gyeonggi Do, Rep. of Korea
{sukjune.yoon, wshan, rex.min, kyung.roh}@samsung.com

Abstract. This paper presents a localization method using circular correlation of omni-directional image. Mobile robot localization, especially in indoor conditions, is a key component in the development of service robots. Though stereo vision is widely used to find location, the performance is limited due to computational complexity and its view angle. To compensate for this, we utilize a single omni-directional camera which can capture 360° panoramic images around a robot at one time. Position of a mobile robot can be estimated by the correlation between CHL (Circular Horizontal Line) of the landmark image and CHL of image captured at the robot position. To accelerate computation, correlation values are calculated based on FFT (Fast Fourier Transform) and to increase reliability, CHLs are warped and correlation values are recalculated. Experimental results and performance in the real home environment show the feasibility of the method.

Keywords: Omni-directional image, Panoramic image, Global localization, Circular correlation, Mobile robot.

1 Introduction

Now days, home and office service robots rapidly have spread into our every day life, such as in the case of clean and surveillance robots. These robots are required to know their position with the help of various sensors. In the case of outdoor environments, GPS (Global Positioning System) can be effectively utilized. However, it can not be used indoors. Therefore, range sensors such as stereo vision, laser range finder and ultra sonic sensor are used to solve the localization problem. Among them, stereo vision systems can handle spatial information at one time at a relatively low cost compared to laser range finders. Stereo vision systems, however, require high computational power and have limited view. To overcome these shortcomings, we utilize the omni-directional camera which can capture 360° panoramic images around a robot at one time.

Spatial resolution of omni-directional cameras is less than that of ordinary cameras because omni-directional camera captures 360° images around robot with the same size CCD. However, omni-directional cameras can be more effectively utilized in the indoor environment, because their view is not limit.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 433 – 441, 2006.

There are two approaches to localize a robot using omni-directional vision systems; by single omni-directional camera or stereo omni-directional cameras. In the latter case, disparity maps which require computational power should be generated [1]. In the former case, landmark images or features are required because position of a robot is calculated with respect to them. Therefore, the former is the more effective indoor application because landmarks can be easily corrected and these landmarks can be easily distinguished between each other due to limited space in the indoor environment.

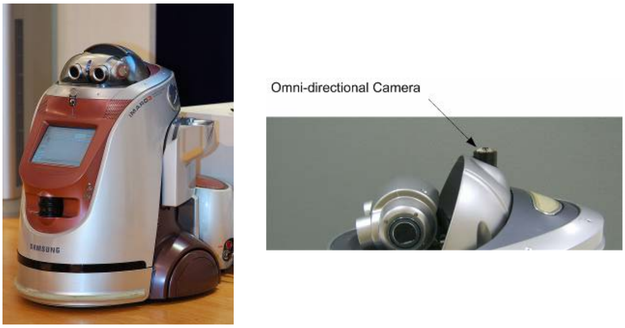
Ishiguro and Tsuji [2] proposed a localization method using Fourier descriptor of omni directional image. However, this method covered only small difference between landmark and current images. Zheng and Tsui [3] first proposed the localization scheme by matching of two panoramic images. In this study, circular dynamic programming and vertical edge properties were used in the matching process. Vertical edges were also used by Yagi *et al.* [4]. Here, they localized a robot by azimuth angles of the vertical edges. However, it is difficult to identify vertical edges exactly in the omni-directional images. Matsui *et al.* [5] proposed localization method using circular correlation of multiple lines in panoramic images. Their method shows good matching performance but requires computational time due to SAD correlation method. Also, it can not compensate for dynamic object. Briggs *et al.* [6] proposed a localization method using SIFT [7] features and dynamic programming. To tolerate occlusion, SIFT features are adopted. Therefore, this method also requires computational time.

In this paper, landmark images are captured at each nodes and the position of a robot is represented by the nearest nodes. These nearest nodes are selected by the circular correlation of CHLs. The more highly correlated CHLs are, the nearer the robot is to the node. To reduce noise effects and compensate for dynamic objects, we propose new fast matching method. This matching process is composed of two stages. First, the best matching nodes are selected by correlation coefficient calculated based on FFT method. To reject false matching and compensate for dynamic objects, final correlation values are recalculated between CHLs of pre-selected nodes and warped current CHLs.

Section II overviews the system including the mobile platform (iMaro) and the omni-directional camera. A new fast correlation calculating method is presented in Section III. In Section IV, the experimental results in the real home environment and performance evaluation are shown. And finally, we conclude and discuss our method in Section V.

2 System Overview

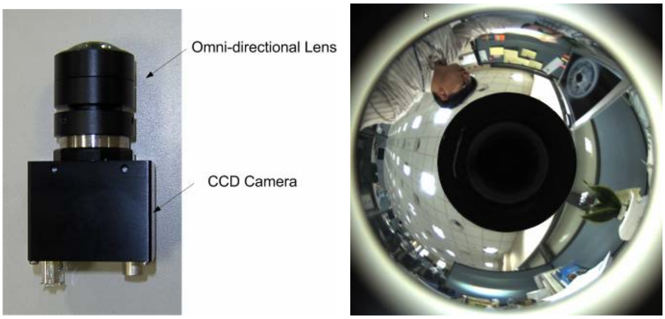
iMaro was developed as an office and home robot at the Mechatronics & Manufacturing Technology Center (Samsung Electronics CO., LTD). This robot, shown in Fig 1, was equipped with a single board computer, motors and controllers, wireless LAN equipment, microphones, a stereo camera, ultrasonic sensors and an IR scanner. To operate 24 hours, a recharge station is located in the working area. Using these sensors, actuators and electrical system, this robot could avoid obstacles, generate the path to destination and find its location w.r.t. its first location. However, it could not solve the kidnap problem. To solve this problem, we utilize omni-directional camera on the top of iMaro, Fig. 1. As mentioned, omni-directional camera can capture 360° images around a robot.



(a) (b)

Fig. 1. Office and home service robot “iMaro”, developed at Mechatronics & Manufacturing Technology Center (Samsung Electronics CO., LTD). The omni-directional camera was installed at top of this robot.

The omni-directional camera and the captured image are shown in Fig. 2. The size of CCD camera is 1/2 inch (1.4 mega pixels). The viewing angle of this camera in Fig 2 (b) is 65° vertically (from 50° to 115°). The effective area is 720 × 720 pixels.



(a) (b)

Fig. 2. Omni-directional camera (a) and the captured image by this camera (b)

3 Global Localization by Circular Correlation

To identify the location w.r.t. nodes, circular correlation between CHLs is used. Before calculating the correlation value, pre-processing should be carried out. Then, the best nodes are selected according to their correlation coefficients. Finally, these best nodes are refined by dynamic warping correlation. The overall process is shown in Fig. 3.

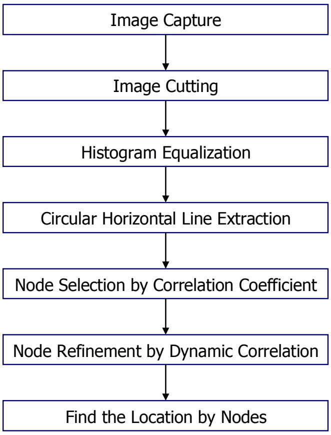


Image Capture
Image Cutting
Histogram Equalization
Circular Horizontal Line Extraction
Node Selection by Correlation Coefficient
Node Refinement by Dynamic Correlation
Find the Location by Nodes

Fig. 3. Overall process flow of global localization

3.1 Pre-processing

In this step, the meaningless black areas in Fig 2 (b) are eliminated. Then, to reduce the effect of illumination condition, histogram equalization is performed. CHL (circular horizontal line) is the line used for correlation computation. Under plane movement assumption, this line always indicates the same plane parallel to ground. Fig. 4 shows the preprocessed images and the CHL in the omni-directional image.

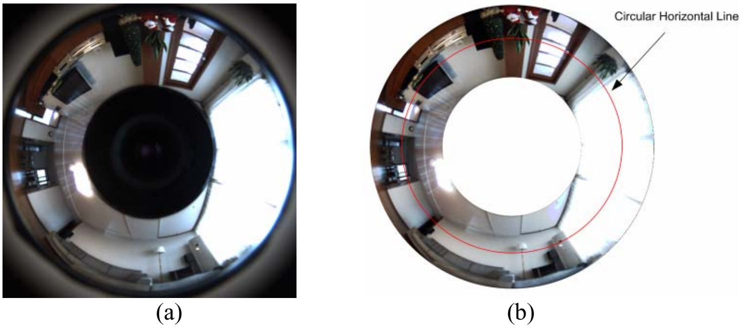


Fig. 4. To reduce the effect of illumination, raw image (a) is processed to (b). CHL (red line) in (b) is used in correlation calculating process.

3.2 Node Selection by Correlation Coefficients Based on FFT

First, the correlation values are calculated by correlation coefficient based on FFT method. Correlation coefficient is described in Eq. (1). The range of this value is from -1 to 1. The closer this value is to 1 or -1, the more correlated the lines are.

$$\rho(\tau) = \frac{C_{xy}(\tau)}{\sqrt{C_{xx}(0) \cdot C_{yy}(0)}} \tag{1}$$

In Eq (1), $C_{xy}(\tau)$ is the cross-correlation between x and y signal. In this case, τ is the rotating angle. $C_{xx}(0)$ and $C_{yy}(0)$ are auto-correlation values. To utilize color information, correlation coefficients for red green and blue are calculated respectively.

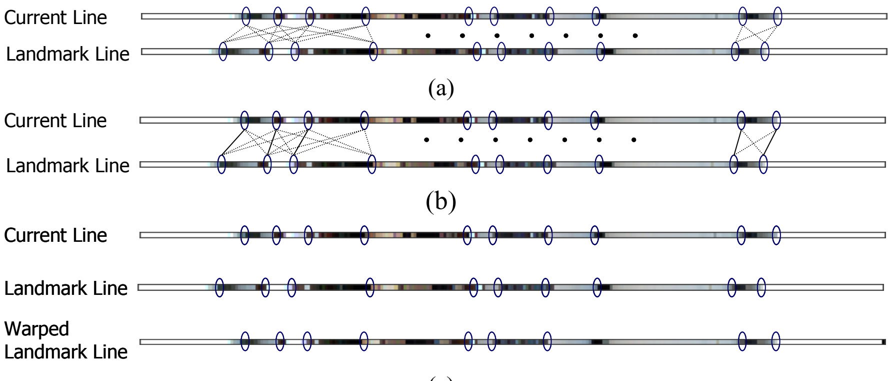
In contrast to Mitsui's method [5], the computational time is very short because the correlation coefficients are calculated based on FFT method. In Mitsui's method [5], *n* times SAD calculation are required to find the correlation value and rotation angle, where *n* is the number of angle step. However, the proposed method finds them in one calculation. Because length of circular line should be to the power of two due to FFT process, the used line size is 256. In this step, several nodes were selected from the sum of correlation coefficients for each color.

3.3 Node Refinement by Dynamic Warping Correlation

In this step, correlation values are re-calculated by dynamic warping. To reduce false selected node and compensate for dynamic objects, the CHL at the selected nodes are warped w.r.t the CHL at the robot position. Then, correlation values between these CHLs are recalculated. The detailed process is described below. In this step, we need not be concerned about the starting point, which is the rotating angle, because the previous step finds the best rotating angle, τ in Eq (1).

1. *Find edges in each CHL, Fig 5(a).* Edges are detected by kernel [-1 0 1].
2. *Match the edges, Fig 4 (b).* Solid lines in Fig. 4 (a) represent matching edges. In this case, maximum allowable distance between matching edges is 35 pixels.
3. *Warp the horizontal circular line at the selected nodes, Fig 4 (c).* The line segments at the selected nodes are stretched or shortened by matching edges.
4. *Compute SAD Correlation Value.* To get correlation value, SAD method, Eq. (2), is adopted.

$$C = \sum_i \left(\left| R_C(i) - R_L(i) \right| + \left| G_C(i) - G_L(i) \right| + \left| B_C(i) - B_L(i) \right| \right) \tag{2}$$



(c)

Fig. 5. Node refinement process. Current circular line (a) is warped to (c) w.r.t. landmark line.

4 Experiment

In this section, the feasibility of the proposed algorithm is presented. To evaluate the proposed global localization method, especially the correlation based landmark image matching process, experiments are performed in a real home environment. First, the mobile robot collected landmark images and saved the CHLs at that location manually. Then, the robot was placed without any information and found the locations by the proposed correlation-based landmark image matching. The performance for error rejection and dynamic object compensation is presented at the last of this section.

4.1 Experimental Condition

Experiments are performed in $102m^2$ apartment, shown in Fig. 6. In this experiment, images are collected from each room. These nodes are $50cm$ apart from each other.

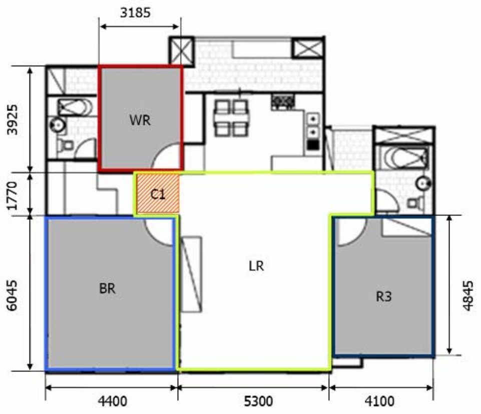
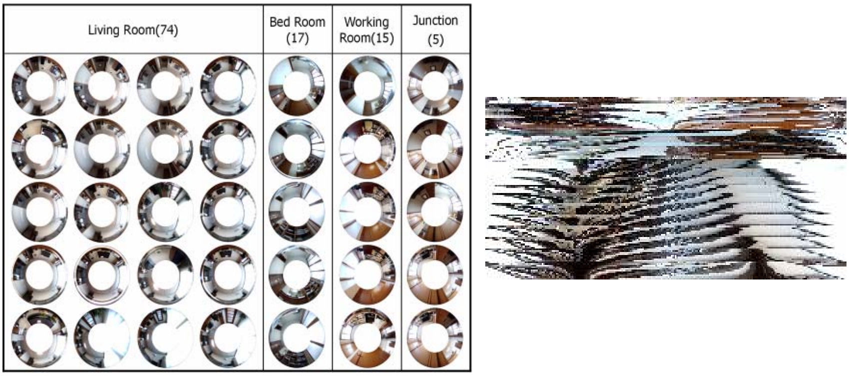


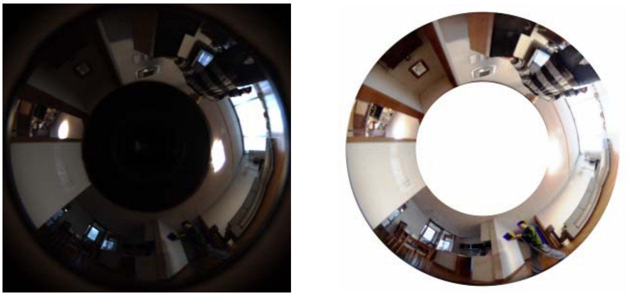
Fig. 6. Experiment environment (Units : *mm*)



(a) (b)

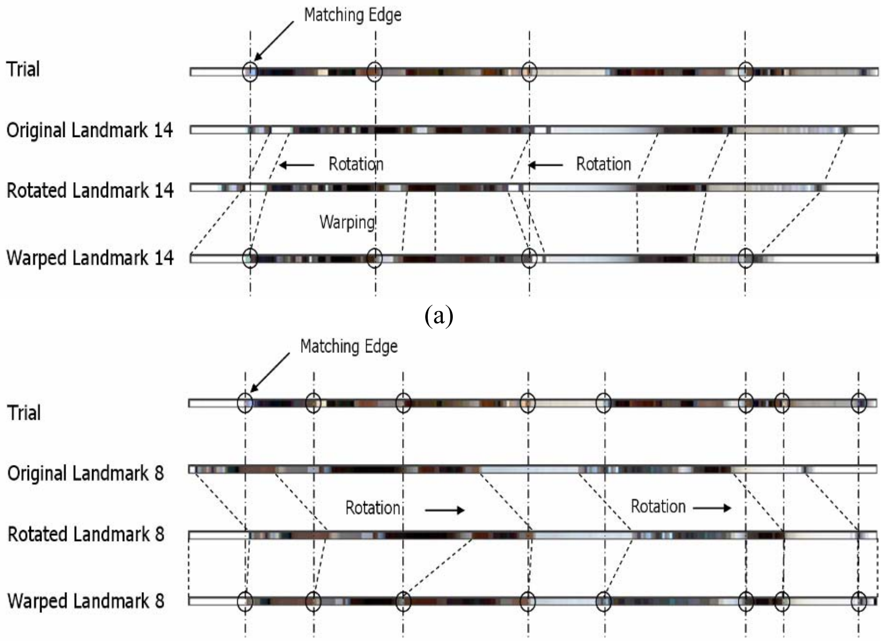
Fig. 7. (a) shows the corrected images and (b) is generated map from these collected images

Fig. 7 shows the collected omni-directional images and the generated CHL map. These images were captured without a moving object. The total number of nodes is 113. Therefore, the size of landmark map is 86784byte (113(Nodes) × 256(Length) × 3(color)).



(a) (b)

Fig. 8. Men were moving in the low light in the (a) and this omni-directional image was pre-processed to (b)



(a)

(b)

Fig. 9. Candidate nodes, (a)-node 14 and (b)-node 8, are selected by node selection process with the rotating angle. These selected landmark CHLs are rotated and warped by the proposed method.

4.2 Experimental Results

To evaluate the performance of dynamic object compensation and mismatching rejection, 20 trials under dynamic environment were carried out. As mention in section 3, 5 best matching landmark images were selected by correlation coefficient and finally 3 best nodes were determined by correlation between CHLs of selected landmarks image and warped CHL of each trial. To evaluate the proposed method, especially the node refinement step, we tried localization under illumination change and dynamic environment, in Fig. 8.

Fig. 8 (a) was captured among node 7, 8, 18 and 19. However, nodes selected by correlation coefficients were 7, 14, 1, 8 and 85 (in order of high correlation) *i.e.* node 14, 1 and 84 were incorrectly select. Then, node refinement was performed. Fig. 9 (a) and (b) shows the process for node 14 and node 8.

In the node selection step, correlation coefficient with landmark image 14 is larger than that with landmark image 8, *i.e.* the position of the robot is predicted closer to node 14. However, this means that matching error existed in this step by the change of illumination and moving people. After the node refinement step, SAD correlation values of current horizontal line with landmark image 14 and landmark image 8 were 55037 and 36303, respectively. This means that the position of the robot is correctly estimated. As shown in Fig 9. (b), the warped CHL at node 8 is closer to the CHL at the robot position than that at node 14. Table 1 shows localization result by nodes for each trial. In table 1, black background cells are the nearest actual nodes, grey cells are the nodes that were close to the nearest nodes and white background cells are

Table 1. The global localization results for 20 trials in the 102m^2 apartment

	Landmark Selection by Correlation Coefficient					Landmark Refinement by Warping		
	1st	2nd	3rd	4th	5th	1st	2nd	3rd
1	60	24	61	31	65	60	65	61
2	63	67	66	60	68	68	67	63
3	70	71	103	72	75	75	71	70
4	50	54	57	56	62	56	62	54
5	30	37	29	51	65	29	37	65
6	25	20	53	18	86	20	25	53
7	21	66	33	39	40	21	39	66
8	3	100	105	2	91	3	2	105
9	7	14	1	8	85	8	7	14
10	47	48	112	95	89	112	95	47
11	11	88	85	89	106	11	89	88
12	112	113	12	90	101	112	113	12
13	77	76	70	71	30	77	76	70
14	79	70	71	72	73	79	70	71
15	87	86	82	100	14	86	100	87
16	89	83	88	75	78	89	83	78
17	52	94	45	64	76	94	76	52
18	96	79	94	111	112	96	94	111
19	106	102	98	100	2	102	106	98
20	107	89	102	103	108	107	108	103

mismatched nodes. As shown in table 1, mismatching nodes of trial 1, 2, 8, 9, 10,17,18,20 were rejected. However, the results of trial 3, 4, 6, 15 got worse. In spite of this result, total localization results were much enhanced.

5 Conclusion

In this paper, we proposed a new fast global localization method by nearest nodes. In the robotic application, though the accuracy of the robot position is important, the localization of the robot without any pre-knowledge is more important. The solution for this problem, called the kidnap problem, is necessary for mobile robots to cope with unexpected accidents. The nearest nodes are determined by correlation of the CHLs (circular horizontal line). Under the plane movement assumption, these lines always indicate always same plane which is parallel to ground.

The proposed algorithm is composed of three steps. At first, to reduce effect of image noise and illumination change, omni-directional images are pre-processed and CHL is extracted. And then, several candidate nodes are selected by correlation coefficient between the CHLs at the nodes and the CHL at robot position. In this step, cross correlations are calculated by FFT. Therefore, compared to Matsui's method [5], the computational speed is very fast. To compensate for various errors, the CHL at the selected nodes are warped and correlation values with warped CHLs are recalculated and finally nodes are determined.

To evaluate the proposed algorithm, experiments were performed in the real home environment. As described in section IV, the proposed algorithm can compensate for the effect of illumination change and dynamic object. Moreover, the required time is reduced by FFT and the update rate is more than 4Hz. From these results, we can conclude that this algorithm can be effective utilized for global localization in the indoor environment in conjunction with other probabilistic approaches, such as particle filtering, condensation and so on.

References

1. Ishiguro H., Tsuji S.: Image-based Memory of Environment. In : IEEE International Conference on Robotics and Automation (1996) 634-639
2. Koyasu H., Miura Jun., Shirai Y.: Mobile Robot Navigation in Dynamic Environments using Omnidirectional Stereo. In : IEEE International Conference on Robotics and Automation (2003) 893-898
3. Zheng J. Y., Tsuji S.: Panoramic Representation for Route Recognition by a Mobile Robot. International Journal of Computer Vision 9(1) (1992) 55-76
4. Yagi Y., Nishizawa Y., Yachida M.: Map-based navigation for a mobile robot with omnidirectional image sensor COPIS. IEEE Transaction on Robotics and Automation 11(5) (1995) 634-648
5. Matsui T., Asoh H., Thompsom S.: Mobile Robot Localization Using Circular Correlations of Panoramic Images. In : IEEE/RSJ International Conference on Intelligent Robots and Systems (2000) 269-274
6. Briggs A., Li Y., Scharstein D., Wilder M.: Robot Navigation Using 1D Panoramic Images. In : IEEE International Conference on Robotics and Automation (2006) 2679-2685
7. Lowe D.G.: Distinctive Image Features from Scale-Invariant Key Points. International Journal of Computer Vision 60(2) (2004) 91-110

Computer-Aided Vision System for MURA-Type Defect Inspection in Liquid Crystal Displays

Hong-Dar Lin[1] and Singa Wang Chiu[2]

[1] Department of Industrial Engineering and Management,
[2] Department of Business Administration,
Chaoyang University of Technology,
168 Jifong E. Rd., Wufong Township, Taichung County 41349, Taiwan
hdlin@cyut.edu.tw

Abstract. This research proposes a new automated visual inspection method to detect MURA-type defects (color non-uniformity regions) on Liquid Crystal Displays (LCD). Owing to their space saving, energy efficiency, and low radiation, LCDs have been widely applied in many high-tech industries. However, MURA-type defects such as screen flaw points and small color variations often exist in LCDs. This research first uses multivariate Hotelling T^2 statistic to integrate different coordinates of color models and constructs a T^2 energy diagram to represent the degree of color variations for selecting suspected defect regions. Then, an Ant Colony based approach that integrates computer vision techniques precisely identifies the flaw point defects in the T^2 energy diagram. The Back Propagation Neural Network model determines the regions of small color variation defects based on the T^2 energy values. Results of experiments on real LCD panel samples demonstrate the effects and practicality of the proposed system.

Keywords: Computer vision system, MURA-type defects; Hotelling T^2 statistics; Ant colony algorithm; Back propagation network.

1 Introduction

Liquid Crystal Displays (LCD) have posed serious threat to Cathode-Ray Tubes (CRT) because the former weigh less, occupy less space, consume less power, possess larger display area, display crisper images, have excellent screen geometry and no refresh-rate flicker, and emit no low-frequency electromagnetic radiation. The above advantages have fueled the wide application of LCDs in high-density products such as digital cameras, personal digital assistants (PDA), cellular phones, etc., but similar popularity has not been gained in color reproduction applications, which have high requirements of color precision and stability.

Various manufacturing techniques and inspection standards are presently used in the display panel industry [1-2]. Accordingly, wide variances of image quality exist among LCD products. Quality control problems such as color non-uniformity defects frequently confront LCD manufacturers [3-4]. But, the color non-uniformity defects are seldom automatically inspected because: difficulties exist in comparing real colors

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 442 – 452, 2006.

and exhibit colors; display brightness spread is unsettled and inestimable; as the production of LCDs involves more innovative and precise manufacturing techniques, stricter specifications are required for LCD surface inspections.

Inspection of color non-uniformity defects by human eyes is as difficult as automatic inspection because wrong judgments are easily made due to human subjectivity and eye fatigues. Therefore, developing an automated visual inspection system will significantly contribute to the quality improvement of LCD products. To meet consumer expectation, monitors should possess trustworthy display properties: maintaining sufficient color uniformity, revealing true colors, and assuring the reproducibility of precision. However, the color uniformity of monitor displays can be easily affected by lighting conditions, environmental factors and subjective human visual perception, so ensuring reliable visual quality of LCDs becomes an important problem to be solved.

Non-uniformity defects are normally called MURA defects [5]. More specifically, MURA is a local lightness variation without a clear contour on a uniformly produced surface, which imparts an unpleasant sensation to human vision [6]. Jiang et al. [7] use luminance measurement equipment to collect data of MURA-type defects and apply analysis of variance and exponentially weighted moving average techniques to develop an automatic inspection procedure. This procedure is limited to 15-inch LCD panels and has difficulties in identifying a defective area crossing two testing blocks. Lu and Tsai [8-9] propose a global approach for automatic visual inspection of micro defects (pinholes, scratches, particles and fingerprints) on TFT panel surfaces using the Singular Value Decomposition (SVD). The SVD is suitable for detecting defects on the patterned TFT-LCD images that contain highly periodical textural structures. Nevertheless, the variety of LCD manufacturing processes and the higher image resolutions of LCD products often lead to less periodical, more complicated textural structures in sensed images.

Since MURA-type defects have no clear contour or contrast, they are very difficult to be detected [6-7]. But, two common MURA-type defects, screen flaw points and small color variations, often exist in LCDs. To detect these defects, this research proposes a new automated visual inspection method based on multivariate Hotelling T^2 statistics, an Ant Colony algorithm, and a Back Propagation Neural network (BPN) model.

2 Proposed Methods

Commonly used in monitoring the mean vector of a multivariate process, the multivariate process control procedure is utilized in this research to locate color variations embedded in LCD displays. We first use multivariate Hotelling T^2 statistics to integrate coordinates of color models and construct a T^2 energy diagram to present the degree of color variations. Then, an Ant Colony based approach that integrates computer vision techniques is applied to detect the flaw point defects in the T^2 energy diagram. The back propagation neural network model is applied to determine the regions of small color variation defects based on the T^2 energy values.

2.1 Multivariate Statistic Applied to Image Models

This research uses color coordinates of the four common color models, RGB, XYZ, Yxy, and L*u*v*, as the multivariate quality characteristics to calculate Hotelling T^2 [10-12] statistics, respectively. Figure 1 shows the calculation procedure of T^2 statistics under the four different color models. The T^2 statistics are described as variations of color uniformity and the T^2 energy diagram presents the degree of different color variations.

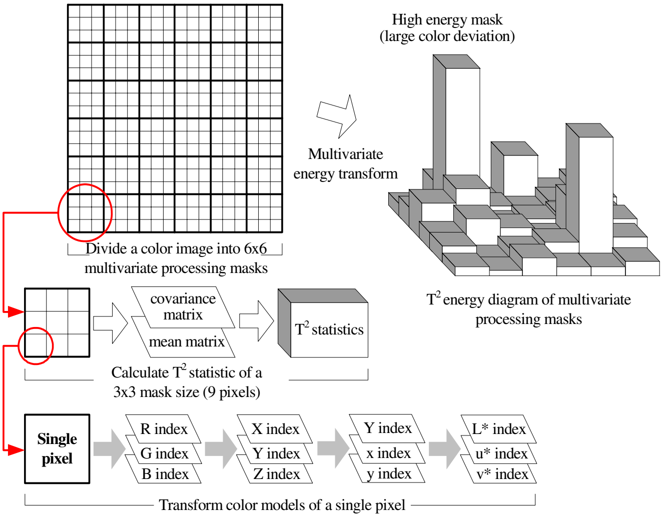


Fig. 1. Calculations of T^2 statistics in four different color models

The values of the T^2 statistics are calculated from many multivariate image processing masks, which are obtained by dividing an image into many image samples with sample size *n* and sample set *m*. For example, an image of 256 x 256 pixels can be divided into 51 x 51 sample sets, each of which has a sample size of 5 x 5 (pixels). The mean matrix ($\bar{X}$) and the covariance matrix (*S*) of the color coordinates in a mask can be written as follows:

$$\bar{X} = \begin{bmatrix} \bar{M}_1 \\ \bar{M}_2 \\ \vdots \\ \bar{M}_k \end{bmatrix} \qquad S = \begin{bmatrix} S^2_{M_1} & S_{M_1M_2} \cdots S_{M_1M_k} \\ & S^2_{M_2} \quad \cdots S_{M_2M_k} \\ & \ddots \\ & & S^2_{M_k} \end{bmatrix} \tag{1}$$

where $\bar{M}_k$ is the average of the color coordinates (*p*) in all pixels of mask *k* (*k=1, …, p*); $S^2_{M_k}$ is the variance of color coordinates (*p*) in all pixels of Mask *k* (*k=1, …, p*); and $S^2_{M_hM_k}$ is the covariance of color coordinates (*p*) among all pixels of masks *k* and *h* (*h=1, …, p, h* $\neq$*k*) (*k=1, …, p*).

The transfer function of a multivariate model for color coordinates can be written as:

$$T^2 = n\left[M_{(i,j)} - \bar{X}\right]^T S^{-1}\left[M_{(i,j)} - \bar{X}\right] \tag{2}$$

where T^2 is the energy value of a multivariate processing mask combining different color coordinates; n is the pixel number of a mask; $M_{(i,j)}$ is the feature matrix of color coordinates; $\bar{X}$ is the mean matrix of image features; and S^{-1} is the covariance matrix of image features. The control limits are as follows:

$$\mathrm{UCL} = \frac{p(m-1)(n-1)}{mn-m-p+1} F_{\psi,\, p,\, (mn-m-p+1)};\ \ \mathrm{LCL} = 0 \tag{3}$$

where F is a tabulated value of the F distribution for the significance level of ψ.

2.2 Ant Colony Algorithm Applied to Flaw Point Defect Detection

This research applies the explorations for the shortest paths of the ant colony algorithm [13-14] to the finding of the maximum T^2 statistics in multivariate processing masks. We combine this concept with computer vision techniques to detect flaw point defects in LCD displays. Figure 2 describes the concept of the proposed ant colony based approach.

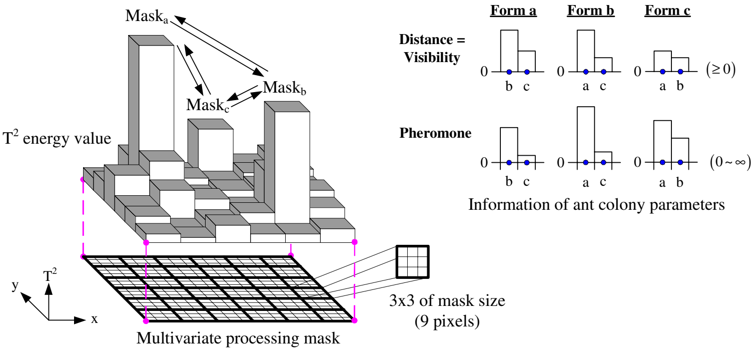


Fig. 2. Concept description of the proposed ant colony based approach applied to flaw point defect detection

Parameter definitions of the proposed model are summarized as follows: O_i is a T^2 value of mask i in T^2 energy diagram; σ_{T^2} is a standard deviation of T^2 values in T^2 energy diagram; $t(O_i)$ is a suspected mask i with a T^2 value of more than $3\sigma_{T^2}$; s_{ij} is a connected path strength between masks i and j; and v_{ij} is a visibility between masks i and j. To proceed with the calculation of the proposed ant colony model, some suspected masks $t(O_i)$ need to be sieved out by using the following equation to reduce the complexity of algorithm computations and increase model practicability:

$$t(O_i) = \begin{matrix} 1 & if\ O_i > 3\sigma_{T^2} \\ 0 & otherwise \end{matrix} \tag{4}$$

After the suspected masks are sieved out, the connected path strength and the visibility between masks *i* and *j* become:

$$s_{ij} = \left[O_i^2 + O_j^2 \right]^{1/2}; \quad i, j = 1, 2, \ldots, n \tag{5}$$

$$v_{ij} = s_{ij}; \quad i, j = 1, 2, \ldots, n \tag{6}$$

The ant colony algorithm could not perform well without pheromone evaporation. Pheromone decay is implemented by introducing a coefficient of evaporation ρ , 0< $\rho \leqq 1$, such that the pheromone quantity from masks *i* to *j* in the *E*-th exploration is:

$$\tau_{ij}(E) = (1-\rho)\ \tau_{ij}(E-1) + \Delta\tau_{ij} \tag{7}$$

The pheromone amount change between masks *i* and *j* is:

$$\Delta\tau_{ij} = \sum_{k=1}^{s} \Delta\tau_{ij}^{k} \tag{8}$$

where the pheromone amount change made by the *k*-th ant is:

$$\Delta\tau_{ij}^{k} = \begin{matrix} \sqrt{t_j} & if\ (i,\ j) \in t(O_i)\ described\ by\ Tabu_k \\ 0 & otherwise \end{matrix} \tag{9}$$

The transition probability from masks *i* to *j* for the *k*-th ant in the *E*-th exploration is:

$$P_{ij}^{k}(E) = \begin{matrix} \dfrac{\left[\tau_{ij}(E)\right]^{\alpha}\left[v_{ij}\right]^{\beta}}{\sum_{k \notin Tabu_k} \left[\tau_{ik}(E)\right]^{\alpha}\left[v_{ik}\right]^{\beta}} & if\ b \notin Tabu_k \\ 0 & otherwise \end{matrix} \tag{10}$$

To locate the positions of flaw point defects, we define *k%* of energy values as the threshold of the masks with higher energy values. If Q_i is greater than the threshold $Q_{\%}$, the mask *i* is judged as a defective mask containing flaw points.

$$Q_i = \sum_{j=1,\ j\neq i}^{n} \tau_{ji}; \quad i = 1, 2, \ldots, n \tag{11}$$

$$Q_{\%} = k\% \sum_{i=1}^{n} Q_i; \tag{12}$$

2.3 Neural Network Model Applied to Small Color Variation Defect Detection

This research uses the back propagation neural network model [15-16] to detect defective regions with small color variations. In the previous section, we eliminate outliers (the flaw point defects) by the ant colony based approach and use the T^2

statistics as the inputs of the neural network model to identify the regions with slight color variations. The T^2 statistics describe the total variations of color uniformity.

The proposed method uses n continuous T^2 statistics in rows of an image as the input values of the BPN model. If the mask size is 5x5 pixels, an image of 256 x 256 pixels will have 2601 T^2 statistics. The totals of the input patterns of the rows are 51/n sets of the T^2 statistics. Each set of the T^2 statistics can be judged as in-control or out-of-control. The selection of the n value directly affects detection performance of the network model. If n is too large, it will increase the calculation load and decrease the detection performance of the network model. On the other hand, if n is too small, it can not locate any defective region in an image. To make network input patterns properly describe the color features of a testing image, this research adopts T^2 statistics in normal images which are inputted into the BPN model for training.

The output layer of the network uses 0 and 1 to represent the in-control and out-of-control decisions. The output result is determined by the criterion of the control limits. If one or more T^2 statistics exceed the control limits, the input pattern is out-of-control. The research uses Hotelling T^2 control limits (Eq. (3)) for the differentiation among n values of T^2 statistics to detect small color variations in LCDs. The data of input patterns must be scaled first. The linear transformation is used to set the range of the input values between [0, 1] to avoid extreme values from affecting the network training results.

Some parameters of the network model, such as learning rate (η), training number, errors, and number of hidden layer nodes, need to be carefully set to achieve good model performance. Uniformly distributed random numbers which range between [-1, 1] are used to set interconnected weights and biased weight vectors (θ) for the training patterns of the model. Sigmoid function is used in the model and the numerical range is between [0, 1].

$$f(x) = \frac{1}{1+e^{-net_j}} \tag{13}$$

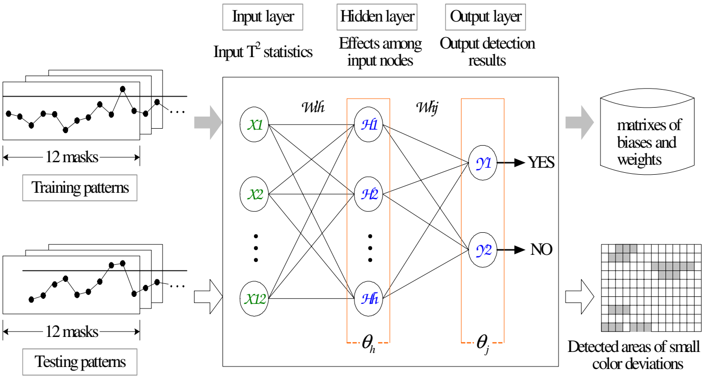


Fig. 3. Network structure of the proposed BPN model

The standard energy function below is used to calculate the variation between expected output and network output.

$$E = \frac{1}{2}\sum_{j}(T_j - Y_j)^2 \tag{14}$$

The stop criterion of the proposed model is based on the proposition of Hush and Horne [17], who use methods of Root Mean Square Error and fixed learning cycles to set the parameters. Figure 3 shows the network structures of the proposed model.

3 Implementation and Performance Evaluation

The experimental environment of this research is properly set up in a dark room so that the captured images can remain steady. We first use a pattern generator to produce a testing pattern, utilize a chromatic CCD to capture the screen images of LCDs, and save the digital signals of the images in a personal computer. Then, we extract color coordinates of the images, transform the coordinates to multivariate T^2 domain, and detect the flaw point defects and small color variation defects. We test 175 images with 256 x 256 pixels in RGB color bands. The mask size is 3 x 3 pixels.

Two kinds of color non-uniformity defects of LCDs are detected in the proposed inspection environment. First, color flaw points are serious color variation defects whose T^2 statistics of masks exceed 3 standard deviations (3σ) of the normal images. The ant colony algorithm is applied to detect the color flaw points. Second, small color variation areas are minor color variation defects whose T^2 statistics of masks lie between 1.5 and 3 standard deviations ($1.5\sigma \sim 3\sigma$) of the normal images. The BPN model is applied to detect the small color variation areas. Currently, the LCD industry inspects these two kinds of defects by human eyes.

In our experiments, two detection phases are planned and conducted for the color variation defects of LCD displays. The first phase implements the proposed ant colony based approach to detect the flaw points. The second phase applies the BPN model to detect the small color variation areas. The color indices of the four color models, RGB, XYZ, Yxy, and L*u*v*, are inputted as the quality characteristics of the multivariate control procedure and the color indices of the images are transformed into T^2 statistics. Training and testing images include pure red, green, and blue images of LCD displays with and without small color variations. To detect the color flaw points of the LCD images, the ant colony method tries to find higher T^2 statistics of suspected masks for locating the positions of flaw points. The proposed ant colony based approach has the following parameter settings: 1) the control value of evaporation = 0.5; 2) the city number = suspected masks; 3) the ant number = 10; 4) α and β of probability calculation equal 1 and 2, respectively; 5) explorations and toleration of stop criteria equal 200 and 0.001, respectively; and 6) threshold parameter $Q_{\%}$ of decision criterion are 0.7%, 1%, 2%, 5%.

To evaluate the performance of the ant colony method in detecting flaw points under different color models, an evaluation index γ is developed and defined as:

$$\gamma = \frac{\text{average of detected flaw points}}{\text{average of erroneous judgements}} = \frac{\#(1-b)}{\#(a+b)} \tag{15}$$

The γ value is a relative ratio, representing the average number of detected flaw points per average number of erroneous judgments. We divide the mask number of erroneously detected defects by the mask number of normal images to obtain Type I error a, and the mask number of missing defects by the mask number of total defects to obtain Type II error b. The larger the value of the index γ, the better the detection result.

As to the detection results of the ant colony method applied to the four color models, the correct detection rates 100(1-a)% are close to 100% and the values of 100(1-b)% approximate 90% when $Q_{\%}$ is set to 0.7%. Table I presents the evaluation indices of the flaw point defect detection under the four different color models. The proposed method performs best when $Q_{\%}$ equals 2% and when the color model L*u*v* is applied based on the criterion of γ value.

Table 1. Evaluation indices of flaw point defect detection under four different color models

Model	Relative ratio	Pure Red	Pure Green	Pure Blue
RGB model	Average of γ	0.32	0.37	1.09
	The best result and γ value	$Q_{\%} = 5\%$ $\gamma = 1.36$	$Q_{\%} = 2\%$ $\gamma = 1.25$	$Q_{\%} = 2\%$ $\gamma = 3.08$
XYZ model	Average of γ	0.32	0.37	1.09
	The best result and γ value	$Q_{\%} = 5\%$ $\gamma = 1.36$	$Q_{\%} = 2\%$ $\gamma = 1.25$	$Q_{\%} = 2\%$ $\gamma = 3.08$
Yxy model	Average of γ	0.36	0.32	0.97
	The best result and γ value	$Q_{\%} = 5\%$ $\gamma = 1.54$	$Q_{\%} = 2\%$ $\gamma = 0.96$	$Q_{\%} = 2\%$ $\gamma = 3.27$
L*u*v* model	Average of γ	0.44	0.40	1.07
	The best result and γ value	$Q_{\%} = 2\%$ $\gamma = 1.47$	$Q_{\%} = 5\%$ $\gamma = 1.11$	$Q_{\%} = 2\%$ $\gamma = 3.36$

In the second phase, the BPN technique is implemented to detect small color variation areas of LCD displays under four different color models. We combine T^2 statistics and the BPN model to establish a color variation detection system for detecting the mean deviations of a multivariate process. The input patterns of the BPN model, each including twelve continuous T^2 statistics, are obtained from the 32 sets of each row of a testing image. The Hotelling T^2 control limit can judge whether the input sets is in-control or out-of-control based on the T^2 statistics of the training outputs. If any T^2 statistic exceeds the control limits, the input set of T^2 statistics is out-of-control.

The parameter settings of the BPN model are: 1) training patterns = two-third of total images; 2) testing patterns = one-third of total images; 3) learning rate = 0.5; 4) iteration cycles = 1,000,000; and 5) error value = 0.01. The testing results of the BPN model can be affected by many factors, such as parameter settings, number of training samples, input patterns of network, mask size, and so on. The index RMSE (Root Mean Square Error) is used to evaluate the performance of the network. As the experimental results show, when the T^2 statistics are close to the control limits, inspection errors will occur because the network has insufficient training cycles.

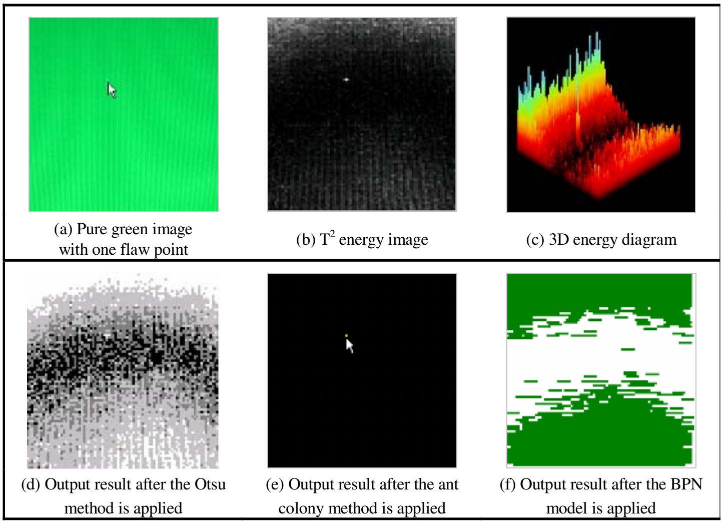
Table 2 shows the evaluation indices of the small color variation defect detection by the BPN method under four different color models. According to the RMSE indices, the proposed method performs best when the network structure 6-4-2 is applied to color model Yxy.

Table 2. Evaluation indices of small color variation defect detection under four different color models

Color Model	**Network Structure**	**RMSE of BPN model**			
		Pure Red	Pure Green	Pure Blue	Average
RGB	6-4-2	0.1325	0.1334	0.1503	0.1387
XYZ	16-9-2	0.1096	0.1519	0.1516	0.1377
Yxy	6-4-2	0.1156	0.1248	0.1503	0.1302
L*u*v*	16-9-2	0.1281	0.1766	0.1837	0.1628

Table 3. Evaluation indices of color non-uniformity defect detection under four different color models

Color model	**Flaw point defect detection**		**Small color variation defect detection**	
	$Q_{\%}$ of the best result	Average of γ values	Network structure	Average of RMSE
RGB	$Q_{\%} = 2\%$	1.5767	6-4-2	0.1387
XYZ	$Q_{\%} = 2\%$	1.5767	16-9-2	0.1377
Yxy	$Q_{\%} = 2\%$	1.5600	6-4-2	0.1302
L*u*v*	$Q_{\%} = 2\%$	1.9733	16-9-2	0.1628


(a) Pure green image with one flaw point

(b) T^2 energy image

(c) 3D energy diagram

(d) Output result after the Otsu method is applied

(e) Output result after the ant colony method is applied

(f) Output result after the BPN model is applied

Fig. 4. Results of a pure green image with a flaw point being processed by the Otsu and the proposed defect detection methods

Table 3 shows the summarized results of the color non-uniformity defect detection by the proposed approaches under four different color models. The ant colony based approach with a $Q_{\%}$ of 2% under the L*u*v* color model performs best in detecting flaw point defects. The BPN method with a network structure 6-4-2 under the Yxy color model performs best in detecting small color variation defects.

Figure 4 partially presents the detection results of a pure green image with one flaw point being processed by the Otsu's method [18] and the two proposed defect detection methods. Apparently, the proposed method performs better than the Otsu's method in the T^2 domain. Both of the flaw point and small color variation regions are correctly detected by our proposed methods.

4 Concluding Remarks

This research proposes new automated visual inspection methods to detect color non-uniformity defects in LCDs. The multivariate Hotelling T^2 statistic is used to integrate different coordinates of the color models and a T^2 energy diagram is constructed to represent the degree of color variations for selecting suspected defect regions. Then, the ant colony based approach identifies the flaw point defects in the T^2 energy diagram, and the BPN model determines the regions of small color variation defects based on the T^2 energy values. As the experimental results indicate, the ant colony method with $Q_{\%}$=2% applied to L*u*v* color model performs the best in detecting flaw point defects based on the γ value. The BPN model with network structure 6-4-2 applied to Yxy color model is the most effective method for detecting small color variation regions based on the RMSE criterion. This research contributes a solution for the detection of MURA-type defects and offers a computer-aided vision system to the LCD panel industry.

Acknowledgments

This study was partially supported by the National Science Council of Taiwan (R.O.C.), Project No. NSC 90-2212-E-324-004 and NSC 91-2212-E-324-004.

References

[1] Kido, T.: In Process Functional Inspection Technique for TFT-LCD Arrays. Journal of the SID, 1 (1993) 429-435

[2] Chen, P.O., Chen, S.H., Su, F.C.: An Effective Method for Evaluating the Image-Sticking Effect of TFT-LCDs by Interpretative Modeling of Optical Measurement. Liquid Crystals, 27 (2000) 965-975

[3] Pratt, W.K., Hawthorne, J.A.: Machine Vision Methods for Automatic Defect Detection in Liquid Crystal Displays. Advanced Imaging, 13 (1998) 52-54

[4] Pratt, W.K., Sawkar, S.S., O'Reilly, K.: Automatic Blemish Detection in Liquid Crystal Flat Panel Displays. SPIE Symposium on Electronic Imaging: Science and Technology, 1998

[5] Lee, Jae Y., Yoo, Suk I.: Automatic Detection of Region-Mura Defect in TFT-LCD. IEICE Transactions on Information and Systems, E87-D, 10 (2004) 2371-2378

[6] Taniguchi, Kazutaka, Ueta, Kunio, Tatsumi, Shoji: A Mura Detection Method. Pattern Recognition, 39 (2006) 1044-1052

[7] Jiang, B.C., Wang, C.C., Liu, H.C.: Liquid Crystal Display Surface Uniformity Defect Inspection Using Analysis of Variance and Exponentially Weighted Moving Average Techniques. International Journal of Production Research, 43, 1 (2005) 67-80

[8] Lu C.J., Tsai, D.M.: Defect Inspection of Patterned Thin Film Transistor-Liquid Crystal Display Panels Using a Fast Sub-image-based Singular Value Decomposition. International Journal of Production Research, 42 (2004) 4331-4351

[9] Lu, C.J., and Tsai, D.M.: Automatic Defect Inspection for LCDs Using Singular Value Decomposition. International Journal of Advanced Manufacturing Technology, 25 (2005) 53-61

[10] Lowry, C.A., Montgomery, D. C.: A Review of Multivariate Control Charts. IIE Transactions, 27 (1995) 800-810

[11] Mason, R.L., Chou, Y.M., Young, J.C.: Applying Hotelling's T^2 Statistic to Batch Process. Journal of Quality Technology, 33 (2001) 466-479

[12] Montgomery, D.C.: Introduction to Statistical Quality Control. 5th edn. John Wiley & Sons, Hoboken, NJ. (2005) 491-504

[13] Dorigo, M., Maniezzo, V., Colorni, A.: Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26 (1996) 29-41

[14] Dorigo, M., Bonabeau E., Theraulaz, G.: Ant Algorithms and Stigmergy. Future Generation Computer System, 16 (2000) 851-871

[15] Kang, B.S., Park, S.C.: Integrated Machine Learning Approaches for Complementing Statistical Process Control Procedures. Decision Support Systems, 29 (2000) 59-72

[16] Smith, A.E.: X-bar and R Control Chart Interpretation Using Neural Computing. International Journal of Production Research, 32 (1994) 309-320

[17] Hush, D.R., Horne, B.G.: Progress in Supervised Neural Networks. IEEE Signal Processing Magazine, January (1993) 8-39

[18] Otsu, N.: A Threshold Selection Method from Gray Level Histograms. IEEE Transactions on Systems, Man and Cybernetics, 9 (1979) 62-66

Effective Detector and Kalman Filter Based Robust Face Tracking System

Chi-Young Seong, Byung-Du Kang, Jong-Ho Kim, and Sang-Kyun Kim

Department of Computer Science, Inje University, Kimhae, 621-749, Korea
{cy1224, deweyman, luckykjh, aiskkim}@gmail.com

Abstract. We present a robust face tracking system from the sequence of video images based on effective detector and Kalman filter. To construct the effective face detector, we extract the face features using the five types of simple Haar-like features. Extracted features are reinterpreted using Principal Component Analysis (PCA), and interpreted principal components are used for Support Vector Machine (SVM) that classifies the faces and non-faces. We trace the moving face with Kalman filter, which uses the static information of the detected faces and the dynamic information of changes between previous and current frames. To make a real-time tracking system, we reduce processing time by adjusting the frequency of face detection. In this experiment, the proposed system showed an average tracking rate of 95.5% and processed at 15 frames per second. This means the system is robust enough to track faces in real-time.

1 Introduction

Real-time face tracking systems have been used in various fields such as access control systems with user identification and user context-aware systems with specific user recognition and tracking. The first step for face tracking is face detection, which locates faces in the video sequence. The second step is tracking the detected face. For the face detection and tracking, various methods have been actively studied to track the faces effectively and in real-time.

Existing tracking methods are mainly based on color, combined color with shape, and face features. Color based approaches track faces by updating the skin color model using Gaussian-Mixture Model[1]. This method does not handle well lighting changes and cases where the skin color exists in the background. To solve these problems, both shape and color information were used[2]. This method used a gradient module to acquire the shape information, and a color module to acquire the color information. The gradient module extracts a sum of intensity gradient around the ellipse's perimeter, and the color module uses a color histogram in the ellipse regions. However, it is difficult to combine the gradient module and the color module on cluttered background images and on images where skin color varies around the ellipse. Face feature based approaches use the spatial relationship of local features of the face such as eyes, nose, and the mouth[3]. This method requires an independent tracker for each feature to track effectively. Another approach uses extended Haar-like features and mutated Dynamic Programming (DP) matching technique[4]. This approach is

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 453 – 462, 2006.

flexible to changes of face angle without needing to retrain the detector. However, face tracking fails where faces do not appear during a given time or the position of the eyes and mouth are out of the matching area.

In this paper, we deal with these problems by using effective detector and Kalman filter. First of all, we construct an effective face detector using PCA[5] and SVM[6] that has acceptable detection speed and is not affected much by the size of the training dataset. As such, it works well with a small quantity of training data. We trace the moving face with Kalman filter that uses static information from the face detector and dynamic information of changes between previous and current frames. To reduce processing time, we adjust the number of trials for face detection to locate faces in the sequence of video frames.

The proposed system performs better face detection because it uses effective features selected from simple Haar-like features with PCA. The SVM classifier, which works well with the binary classification of faces and non-faces, also contributes to better face detection. The Kalman filter, which has the best prediction ability, makes it possible for the system to trace faces efficiently with the face detector.

2 Overview of Face Tracking System

Fig. 1 shows the main structure of our face tracking system. The first step is to detect the faces in the input sequence with PCA and SVM. The second step is to extract changes between previous and current frames. The last step is to combine the static and dynamic information, which is then used for the Kalman filter.

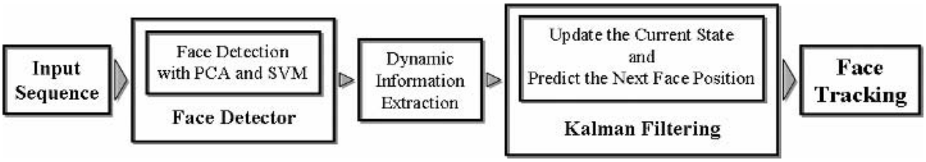

Input Sequence
Face Detection with PCA and SVM
Face Detector
Dynamic Information Extraction
Update the Current State and Predict the Next Face Position
Kalman Filtering
Face Tracking

Fig. 1. Main structure for face tracking

3 Face Detector Construction

To increase the face tracking rate from an image sequence, the detector must guarantee a high detection rate. Fig. 2 shows the structure of a face detector using PCA and SVM. The data collector accumulates the value of Haar-like features. In the second step, the feature space is transformed into the space of the principal components. The selected features from the principal component space are used as feature vectors of the SVM. In the next step, the SVM classifier is trained with the training patterns. In the last step, the SVM classifier classifies regions into faces and non-faces.

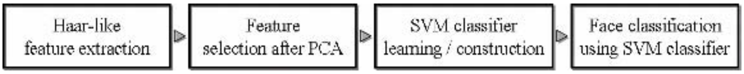

Haar-like feature extraction
Feature selection after PCA
SVM classifier learning / construction
Face classification using SVM classifier

Fig. 2. Four steps of the face detection

3.1 Feature Extraction

The face detector is based on the simple rectangle features presented by Viola and Jones[7]. They measure the differences between region averages at various scales, orientations, and aspect ratios. The rectangle features can be evaluated extremely rapidly at any scale (see Fig. 3). However, these features require very large training datasets. Therefore, after analyzing principal components, we select useful features from each of five rectangle features. These selected features are used as feature vector for the SVM. Experiments demonstrate that they provide useful information and improve performance of accurate classification with small training datasets.

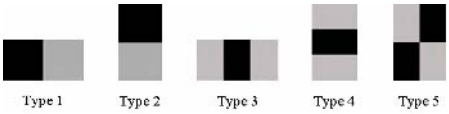


Fig. 3. Used Haar-like features

We used 12 principal components that explain features with more than the cumulative explanation rate of 90%. From the 12 principal components, we selected 288 useful features from all possible 162,336 Haar-like simple features. Fig. 4 shows the 288 useful features selected using PCA. Consequently, a training image is converted to 288 values corresponding to the useful features, and our SVM classifier uses this input vector of 288 dimensions for training.

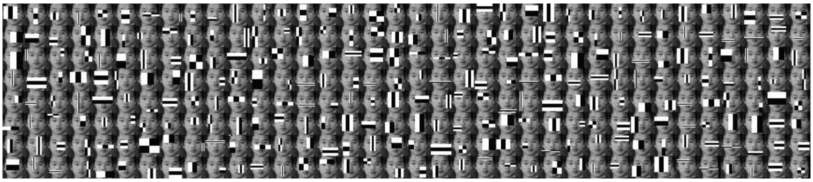


Fig. 4. The 288 useful features selected

3.2 Training of Classifier

The training data was 1000 face and 1000 non-face images randomly chosen from the MIT CBCL Face Data Set[8]. Each image was normalized to 24×24 pixels. Fig. 5 shows part of the training data that consists of face and non-face images.

(a) Face data

(b) Non-face data

Fig. 5. Parts of training data

Fig. 6 gives an overview of detection progress, which includes simple feature extraction, feature analysis, and classifier construction. Firstly, from the Haar-like simple features, useful features are selected using PCA. Training images, as shown in Fig. 5, are converted to input vectors of 288 dimensions with the selected features. The SVM classifier uses these input vectors for training. Fig. 7 shows examples of face detection using our face detector with the CMU test set.

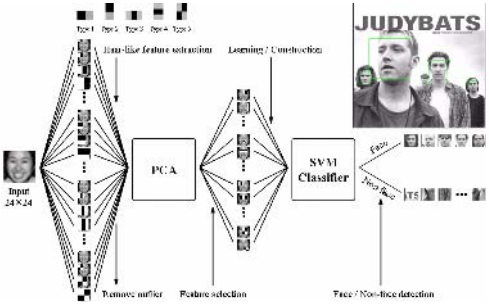


Fig. 6. The basic structure for face detection

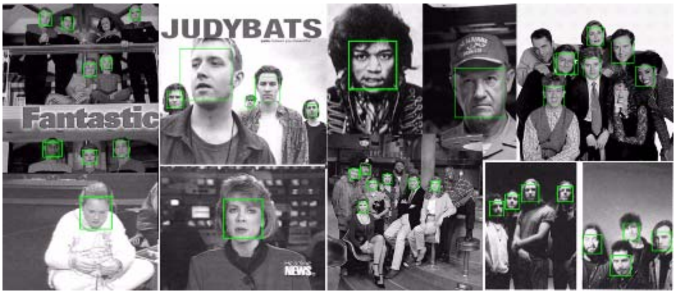


Fig. 7. Detection results derived by applying our face detector with some of the CMU test set

4 Face Tracking with Kalman Filter

The Kalman filter[9] is a well known method in field of motion prediction. It provides optimal prediction on a linear dynamic system that has white Gaussian noise. Also, it is easy to implement with low computational cost, because it is a successive and recursive algorithm. As shown in Fig. 8, the Kalman filter consists of two states: prediction of the next state and update of the current state. Each state is computed recursively and iteratively.

In this paper, we used the Kalman filter in order to reduce the cost of operation and improve the tracking rate in the sequence of video images. The state vector of the Kalman filter uses static information from the face detector and dynamic information between frames.

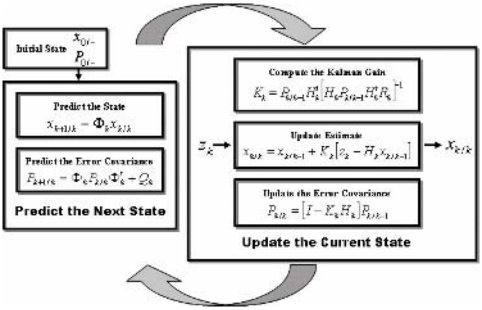

Initial State $x_{0/-}$ $P_{0/-}$
Predict the State
$x_{k+1/k} = \Phi_k x_{k/k}$
Predict the Error Covariance
$P_{k+1/k} = \Phi_k P_{k/k} \Phi_k^t + Q_k$
Predict the Next State
Compute the Kalman Gain
$K_k = P_{k/k-1} H_k^t \left[H_k P_{k/k-1} H_k^t R_k \right]^{-1}$
z_k
Update Estimate
$x_{k/k} = x_{k/k-1} + K_k \left[z_k - H_k x_{k/k-1} \right]$
$x_{k/k}$
Update the Error Covariance
$P_{k/k} = \left[I - K_k H_k \right] P_{k/k-1}$
Update the Current State

Fig. 8. Cycle routine of the Kalman filter

4.1 Face Tracking Modeling

For efficient face tracking, the Kamlan filter requires a setting of an appropriate trace model. We set the state vector as the center coordinates (x, y) of the detected faces and the quantity of change $(\Delta x, \Delta y)$ between previous and current frames.

The state vector of the Kalman filter in time t is defined as:

$$\mathbf{x}(t) = [x, y, \Delta x, \Delta y]^T, \tag{1}$$

The Kalman filter assumes that the system state vector, $x(t)$, evolves according to time as:

$$\mathbf{x}(t+1) = \Phi(t)\mathbf{x}(t) + w(t) \tag{2}$$

where $w(t)$ is a zero mean Gaussian noise with covariance $Q(t)$.

The measurement vector is given by:

$$\mathbf{z}(t) = H(t)\mathbf{x}(t) + v(t) \tag{3}$$

where $v(t)$ is another zero mean Gaussian noise factor, with covariance $R(t)$.

We assumed that faces move with uniform speed and linear direction. The state transition matrix $\Phi(t)$ is defined as follows:

$$\Phi(t) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The input vector is a four dimensional vector that uses the center coordinates of the face and changes of the x, y axis. Therefore, the measurement matrix $H(t)$ is defined as:

$$H(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{5}$$

5 Experimental Results

The proposed face tracking system was developed using Visual C++ on an 2.4GHz P4 PC with the Microsoft Windows operating system. To evaluate our system, we experimented with face detection and tracking on various video sequences. The video sequences were collected from various sources such as the Open-Video web site[10], captured video from TV broadcasts, Boston University's IVC Head Tracking Video Set[11], and PC-cameras.

5.1 Tracking Performance by Adjusting the Frequency of Face Detection

Table 1 shows the accuracy rate of face detection and tracking by frequency of face detection. Fig. 9 shows the trajectory for the x position of a tracked face for each frequency of face detection in Table 1.

Table 1. Result of face detection and tracking by adjusting the frequency of detector

Frequency of detection	Face detection		Face tracking	
	False detects	Detection (%)	False tracks	Tracking (%)
(a) Every frame	261/2490	89.5	129/2490	94.9
(b) 2nd frame	120/1245	90.4	134/2490	94.6
(c) 3rd frame	85/830	89.8	433/2490	82.6

Fig.9 (a) is the traced result with face detection for every frame. (b) is the traced result for face detection every second frame. Although detection is performed every second frame, it showed a similar result to that for every frame. (c) is the traced result for face detection for every third frame. In this case, the accuracy of the tracking rate is lower than (a) and (b), and showed an unstable trajectory.

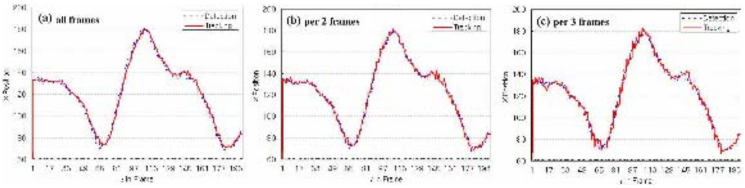


Fig. 9. Trajectory for the x position of the tracked face according to the frequency of face detection

Fig. 10 shows the trace result for face detection every second frame. Although the face detector performs detection every second frame, we can trace faces successfully, as shown in Fig. 10. Because this results in a 50% reduction of face detection time, we get an improvement of system performance.

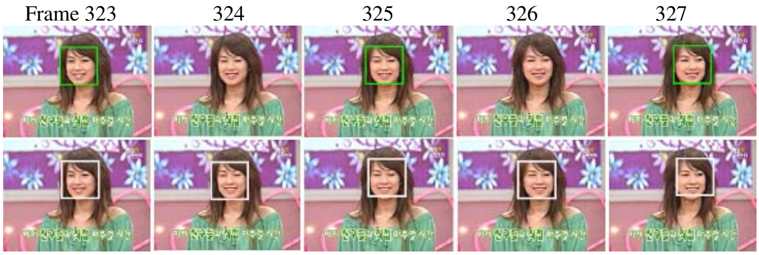


Fig. 10. Results of detection and tracking every 2nd frame

5.2 Experiments on Various Image Sequences

Fig. 11 and 12 shows face detection and tracking results on sequences under various conditions. The first rows of each figure show the results of face detection with PCA and SVM, and the second rows show face tracking with the Kalman filter. As shown in Fig. 11, although face detection fails due to rotated faces not being included in the training of the face detector, face tracking is possible. Fig.12 (a) shows successful face tracking where occlusion occurs locally in the face region (frames 101 and 103). In Fig.12 (b), the face detector extracts two regions as faces incorrectly. However, faces were continuously traced by the Kalman filter (frames 419 and 473).

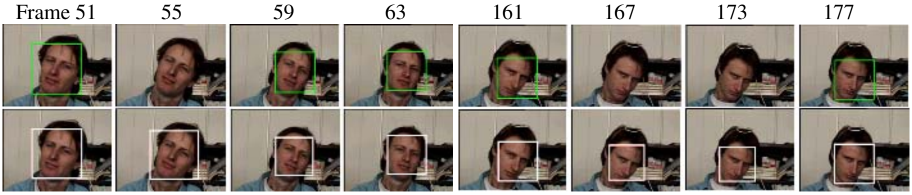
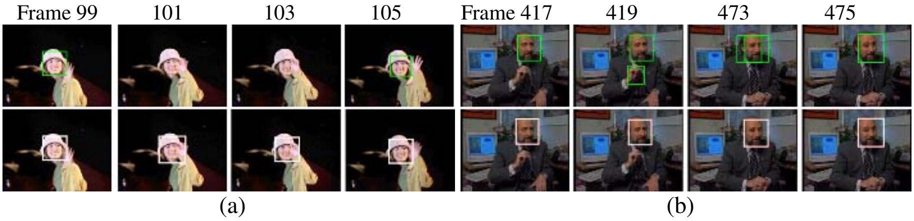


Fig. 11. Results of face tracking in a heavily rotated face sequence



(a)

(b)

Fig. 12. Result of face tracking with local occlusion in the face region or spurious detection

Table 2 presents the face detection and tracking rates on image sequences under various conditions. The face detector with PCA and SVM shows an average detection rate of 89.5%. The tracking system shows an average of 95.5% of high tracking efficiency. This is due to the high detection rate of the face detector and the prediction ability of the Kalman filter.

Table 2. Comparison of detection and tracking rates in various sequence of images

Sequence	Frame	Detection fail	Detection (%)	Tracking fail	Tracking (%)
1	2490	261	89.5	134	94.6
2	730	74	89.8	26	96.4
3	200	20	90.2	4	97.8
4	200	33	83.6	11	94.4
5	620	44	92.9	17	97.3
6	1150	133	88.4	116	89.9
7	820	69	91.6	25	96.9
8	500	52	89.7	16	96.8
Total	**6710**	**686**	**89.5**	**351**	**95.5**

5.3 Comparison with Related Works

To compare our tracking system with related works, we selected method[4] among the various tracking methods. This method uses Viola-Jones[5]'s basic features and additional Lienhart[12]'s extended Haar-like features to increase the detection rate on various face poses. It creates a deformable face graph of one dimension after subdividing eyes and mouth in the detected face. When it fails to detect the face, it matches the face graph of previous and current frames using DP to trace faces continuously.

The sequences[11] used for comparison are the same as those used in method[4]. The input sequence has up, down, left, and right pose variations of faces and frequent light changes. We experimented on 2000 frames from 10 sequences, and the results are given in Table 3.

Table 3. The comparison results of our system and method[4]

	Method[4]				**Our Approach**			
Sequence	**D.F.**	**D.R. (%)**	**T.F.**	**T.R. (%)**	**D.F.**	**D.R. (%)**	**T.F.**	**T.R. (%)**
1 (jam5.avi)	33	83.5	0	100	12	94.0	0	100
2 (jary.avi)	38	81.0	0	100	33	83.6	11	94.4
3 (jim2l.avi)	47	76.5	16	92.0	20	90.2	4	97.8
4 (llrx.avi)	104	48.0	40	80.0	48	75.8	21	89.6
5 (llm1.avi)	30	85.0	0	100	14	93.0	0	100
6 (vam7.avi)	76	62.0	24	88.0	35	82.6	9	95.5
7 (jam9.avi)	84	58.0	12	94.0	37	81.4	6	97.2
8 (llm1r.avi)	156	22.0	5	97.5	50	75.0	14	93.2
9 (llm4.avi)	90	55.0	20	90.0	24	87.8	11	94.6
10 (mll6.avi)	31	84.5	0	100	8	96.0	0	100
Total	**689**	**65.6**	**117**	**94.2**	**281**	**85.9**	**75**	**96.2**

(D.F.: Detection fail, D.R.: Detection rate, T.F.: Tracking fail, T.R.: Tracking rate)

For sequences #2 and #8, our system showed lower tracking performance, as shown in Fig. 13. Our system missed faces because of extreme variations in pose and low illumination.

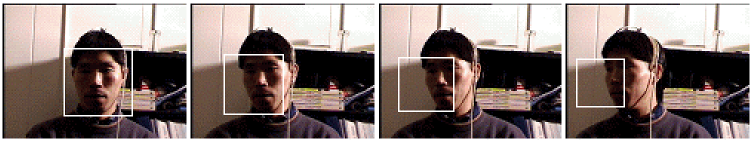


Fig. 13. Result for sequence #8

For sequences #5 and #10, which had lower variations in pose and good illumination, both methods showed a tracking rate of 100%, as shown in Fig. 14.

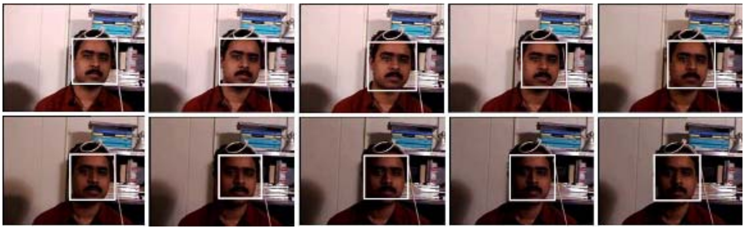
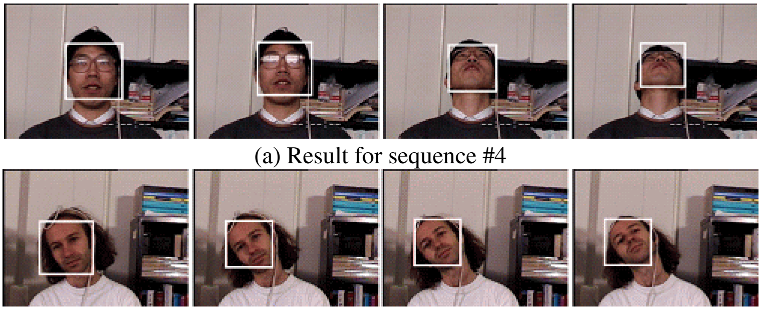


Fig. 14. Result for sequence #5

However, for the sequences with up, down, left, and right face movements, such as #7 and #9, our system showed a higher tracking rate generally. Especially, our system was much better than method[4] where the sequence has a fast moving face (sequences #1 and #3), the front of the face does not appear during a given time, and the face is rotated extremely (sequences #4 and #6), as shown in Fig. 15.



(a) Result for sequence #4

(b) Result for sequence #6

Fig. 15. Tracking result for sequences #4 and #6

6 Conclusion

We proposed a robust face tracking system based on effective face detector and Kalman filter to trace faces in real-time on video sequences having various poses, illumination, and movement.

Firstly, we designed an effective face detector with PCA and SVM. Useful features to discriminate between faces and backgrounds are extracted from simple Haar-like features with PCA. The feature vectors are used for learning patterns for the SVM that is appropriate for binary classification. The Kalman filter for tracking uses the face position of each frame, which is the result of the face detector and changes in face position between frames, as parameters for the state vector. The prediction process of the Kalman filter presents optimal face position prediction for the next frame. Consequently, we implemented a tracking system combining a face detector with a high detection rate and the prediction ability of Kalman filter to get synergy.

In the experiment, we obtained an average tracking rate of 95.5% at 15 frames per second on sequences of 320×240 pixel images. This suggests the system is robust enough to track faces in real-time. In the future, we will strengthen the system to be able to track faces in extreme poses and under significant lighting changes.

References

1. Schwerdt, K. and Crowley, J.L.: Robust Face Tracking Using Colour. IEEE Int'l Conf. Automatic Face and Gesture Recognition, (2000) 90-95
2. Birchfield, S.: Elliptical Head Tracking using Intensity Gradient and Color Histograms. IEEE Computer Vision and Pattern Recognition, (1998) 232-237
3. Hager, G. and Toyama, K.: X Vision: A Portable Substrate for Real-Time Vision Applications. Computer Vision and Image Understanding, Vol. 69, No. 1, (1998) 23-37
4. Yao, Z. and Li, H.: Tracking a Detected Face with Dynamic Programming. Image and Vision Computing, Vol. 24, No. 6, (2006) 573-580
5. Johnson, R.A. and Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice Hall, (2002) 356-395
6. Vapnik, V.: The Nature of Statistical Learning Theory 2nd Edition. Springer, (2001)
7. Viola, P. and Jones, M.J.: Robust Real-Time Face Detection. International Journal of Computer Vision, Vol. 57, No. 2, (2004) 137-154
8. MIT CBCL - Face Database, http://www.ai.mit.edu/projects/cbcl/
9. Welch, G. and Bishop, G.: An Introduction to the Kalman filter. University of North Carolina at Chapel Hill, Department of Computer Science, TR 95-041, (2004)
10. Open-Video, http://www.open-video.com
11. Boston University IVC Head Tracking Video Set, http://www.cs.bu.edu/groups/ivc/
12. Lienhart, R. and Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. IEEE Int'l Conf. Image Processing, Vol. 1, (2002) 900-903

Recognizing Multiple Billboard Advertisements in Videos

Naoyuki Ichimura

National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba Central 2, 1-1-1, Umezono, Tsukuba, Ibaraki 305-8568, Japan
nic@ni.aist.go.jp
http://staff.aist.go.jp/naoyuki.ichimura/

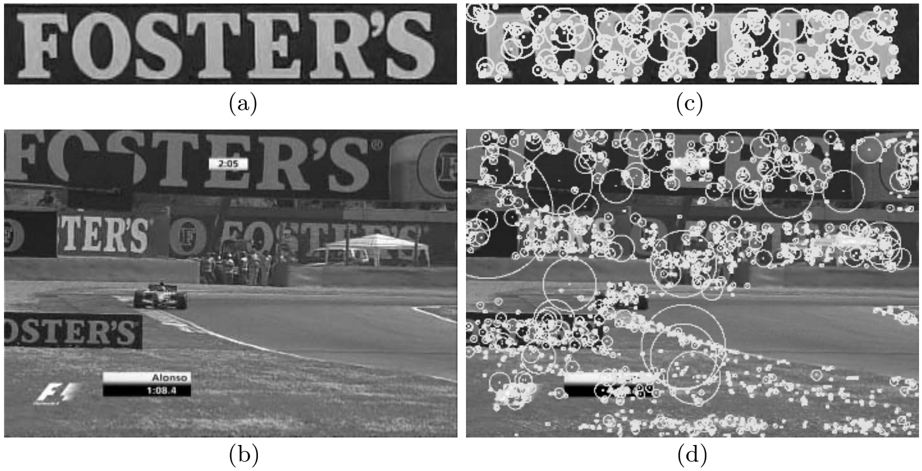
Abstract. The sponsors for events such as motor sports can install billboard advertisements at event sites in return for investments. Checking how ads appear in a broadcast is important to confirm the effectiveness of investments and recognizing ads in videos is required to make the check automatic. This paper presents a method for recognizing multiple ads. After obtaining point correspondences between a model image and a scene image using local invariants features, we separate the point correspondences of an instance of an ad by calculating a homography using RANSAC. To make the use of RANSAC feasible, we develop two techniques. First, we use the ratio of distances of descriptors to reject outliers and introduce a novel scheme to set a threshold for the ratio of distances. Second, we incorporate an evaluation on appearances of ads into RANSAC to reject the homographies corresponding to appearances of ads which are never observed in actual scenes. The detail of a recognition algorithm based on these techniques is shown. We conclude with experiments that demonstrate recognition of multiple ads in videos.

1 Introduction

The sponsors for events such as motor sports can install billboard advertisements at event sites in return for investments. Checking the positions and areas of ads in a broadcast is important to confirm the effectiveness of investments and recognizing ads in videos is required to make the check automatic. Figure 1 (a) and (b) show an example of the recognition problem. Given a model image of an ad shown in Fig. 1(a), we try to calculate the positions and areas of the instances of the ad in a scene image shown in Fig. 1(b). An issue in recognition is that ads in videos could have various appearances depending on their sizes and sites, the position, angle and zoom of a camera, and other factors. In the scene image, the changes in scales and intensities of the four instances of the ad are observed as well as occlusions. As this example demonstrates, we need to cope with deformations, illumination changes and occlusions in recognition.

Using local invariant features is a way to develop a recognition algorithm which has tolerance to deformations, illumination changes and occlusion. Local invariant features are constructed by making the following two components invariant to deformations and illumination changes: (i) local region detectors,

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 463–473, 2006.



(a) (c)

(b) (d)

Fig. 1. An example of a recognition problem. (a) A model image of an ad. (b) A scene image obtained from a broadcast of a car race (F1). Our aim is to find the positions and areas of the instances of the ad in the scene. In the scene, the changes in scales and intensities of the four instances of the ad are observed as well as occlusions. As this example demonstrates, we need to cope with deformations, illumination changes and occlusions of ads in recognition. (c),(d) Examples of local regions. The circles represent local regions in which descriptors are computed. Only 33% of all regions are shown for the sake of clarity. Local invariant features calculated in the local regions are used to develop a recognition algorithm which has tolerance to deformations, illumination changes and occlusions of ads.

(ii) descriptors describing local regions. The circles in Fig. 1 (c) and (d) represent the examples of local regions. Using multiple local regions as shown in these figures, we can recognize ads even if they are partly occluded because the features of visible portions are available. Many different techniques for detecting and describing local regions have been developed. Local region detectors are based on interest points extraction in scale space [1,2,3,4,5,6,7,8,9,10] , region segmentation [5,11,12,13], edge detection [13,14] etc. Descriptors are derivatives of intensities [2,8], image patches obtained by region normalization [4], moment features [11,12,13], wavelet coefficients [9], histograms of gradient orientations [3,5,6,7,10,14] etc. The local invariant features constructed from these detectors and descriptors can be invariant to various transformations of a local region shape and an intensity such as similarity and affine transformations. Therefore point correspondences between a model image and a scene image are found in spite of deformations, illumination changes and occlusions.

We, however, encounter another issue after point correspondences are obtained. The issue is separation of the point correspondences of a single instance of an ad. If there are multiple instances of an ad in a scene, point correspondences of instances are mixed and the separation of them is indispensable to recognize an instance. Although many methods for object matching based on local invariant features have been proposed [2,3,4,5,7,14], the problem of multiple instances rarely gets much attention so far.

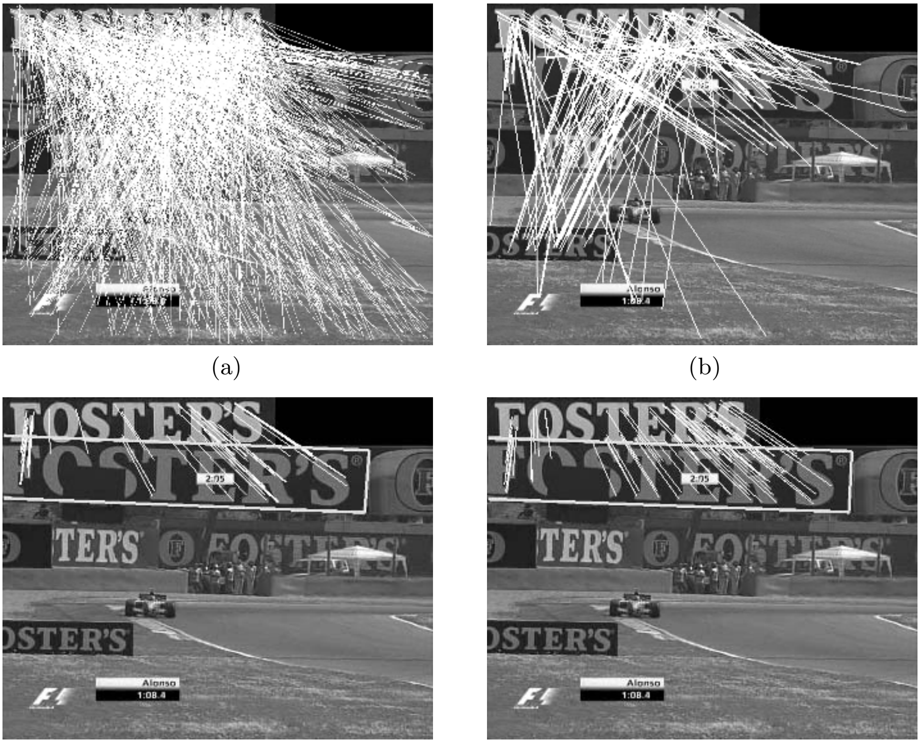
The purpose of this paper is to develop an algorithm for recognizing multiple instances of ads. Basically, the segmentation problem to separate an instance can be treated as a model fitting problem for point correspondences with outliers (false matches) [15]. We can use a homography [16] as the general model which gives a global constraint for grouping the point correspondences of a single instance because billboards are planes in most cases. Thus to extract the point correspondences that obey a homography is equivalent to separate the point correspondences of an instance. Although a robust estimator, RANSAC (RANdom SAmple Consensus) [17], can be used to calculate a homography while rejecting outliers, it would fail if the ratio between inliers and outliers is low as noted in [7,13]. An alternative to RANSAC is Hough transformation [7], but it needs an approximation of a homography such as a similarity transformation to reduce the number of dimensions of a voting space.

In order to take advantage of a homography as the general model, we introduce two techniques for model fitting by RANSAC. First, we use the ratio of distances of descriptors to reject outliers [7,9] and introduce a novel scheme to set a threshold for the ratio of distances. The scheme ensures the reasonable number of point correspondences so that we can use RANSAC to calculate a homography. Second, we incorporate an evaluation on appearances of ads into RANSAC. Using the evaluation, we can reject the homographies corresponding to appearances of ads which are never observed in actual scenes. The evaluation is useful to reject the homographies computed from samples containing outliers that coincidentally yield a reasonable amount of votes. We will show the detail of an algorithm based on these techniques in the following sections and demonstrate by experiments that our method recognizes multiple instances of ads in various situations even if only one model image is given.

2 Recognition Algorithm

We show the proposed recognition algorithm using the following notations: The local invariant features of a scene image are represented by a set $\boldsymbol{f}_i^s = \{\boldsymbol{p}_i^s, \sigma_i^s, \boldsymbol{d}_i^s\}$, where, $\boldsymbol{p}_i^s$ indicates the position of a local region expressed in homogeneous coordinates, σ_i^s the scale in which the local region is found, $\boldsymbol{d}_i^s$ the descriptor and i index. Similarly, the local invariant features of a model image are denoted as $\boldsymbol{f}_j^m = \{\boldsymbol{p}_j^m, \sigma_j^m, \boldsymbol{d}_j^m\}$. The distance between the features is measured by the Euclidean distance between descriptors, $d_{ij} = \left\|\boldsymbol{d}_i^s - \boldsymbol{d}_j^m\right\|$.

We use Hessian-Laplace detector [7,8] to detect local regions. The detector is selected based on the results of the preliminary test of 3 local region detectors, Harris-Laplace detector [18], Hessian-Laplace detector and Difference of Gaussian (DoG) detector [3,7], using several images. The radii of circular local regions are determined as $20\sigma_i^s$ and $20\sigma_j^m$. As a descriptor, we use the gradient location-orientation histogram (GLOH) [19] based on the fact that its high invariance has been shown in the comparative experiments of [19]. For gradient



(a)

(b)

(c)

(d)

Fig. 2. An example of the recognition process. In (c) and (d), only inliers are shown as point correspondences. The results of image alignment are expressed in rectangles. (a) Matching result by the nearest neighbor method. Only 20% of all matches are shown for the sake of clarity. (b) Putative matching using Eqs.(1) and (2). Compared to (a), false matches have been reduced while keeping the inliers. However, point correspondences of the multiple instances of the ad are mixed. (c) Putative alignment by a homography. The position of the single instance is obtained by RANSAC in which an evaluation on appearances of ads is incorporated. (d) Guided matching and final alignment results. The search regions are restricted based on the results in (c), and point correspondences are obtained only from regions around the instance. Compared to Fig. 2 (c), a homography is calculated with more inliers which lead to final alignment of the instance. Regardless of the existence of the multiple instances, the point correspondences of the single instance have been separated.

locations, we use 4 bins in radial direction and 8 bins in angular direction, which results in 25 location bins. Gradient orientations are represented by 16 bins. The number of dimensions of a descriptor is 25 × 16 = 400.

2.1 Putative Matching with Rejection of False Matches

The nearest neighborhood method is adopted to find matches between a model image and a scene image. The feature $\boldsymbol{f}^{m}_{j_{1NN}}$ with the index of $j_{1NN} = \arg\min_j d_{ij}$

is matched with the feature $\boldsymbol{f}_i^s$. Figure 2 (a) shows an example of matches. As shown in this figure, there are many false matches mainly due to a background. Such false matches make the ratio between inliers and outliers low.

To reduce the number of false matches, only the point correspondences that satisfy the following equation are extracted [7,9]:

$$d_{ij_{1NN}}/d_{ij_{2NN}} < t,\ 0 \leq t \leq 1\,, \tag{1}$$

where, j_{2NN} is an index for the second nearest neighbor and t is a threshold value. The number of point correspondences extracted using Eq. (1) increases as t increases, and it reaches the maximum in the nearest neighborhood method corresponding to $t = 1$. Since the relationship between $d_{ij_{1NN}}$ and $d_{ij_{2NN}}$ depends on scene images, it is difficult to estimate the number of extracted point correspondences if we use a fixed threshold value as in [7,9]. If the number of point correspondences is too small, 4 or more inliers needed to calculate a homography may not be included. If it is too large, the ratio between inliers and outliers could be low. We actually had difficulty to set an appropriate threshold for many scenes.

To ensure the reasonable number of point correspondences, we increase t according to the following equation until the number of extracted point correspondences reaches a certain number P_{min}:

$$\begin{aligned} &t\,(k+1) = \alpha t\,(k)\ , \\ &\alpha = 1.01,\ t\,(0) = 0.80,\ k = 0, 1, 2, \ldots\ , \end{aligned} \tag{2}$$

where, k is the number of iterations, and α is the coefficient to control increase in t. By this scheme which gradually increases t, P_{min} point correspondences are ensured while trying to get the high ratio between inliers and outliers as possible as we can. Thus this scheme enable us to use RANSAC to calculate a homography. We found empirically that $P_{min} = 60$ is a good choice for many scenes. If P_{min} point correspondences cannot be extracted, our algorithm decides that there is no ad in a scene.

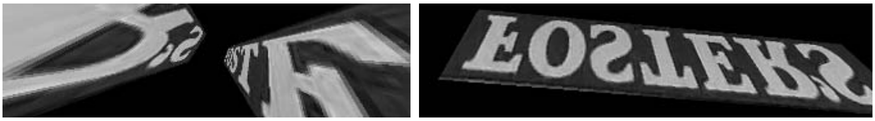
Figure 2 (b) shows an example of outlier rejection based on Eqs. (1) and (2). Compared to Fig. 2 (a), in which the nearest neighbor method is used, false matches have been reduced while keeping the inliers.

Although false matches can be reduced, the point correspondences of the instances of the ad are mixed in Fig. 2 (b). Because the point correspondences of other instances work as false matches in calculation of the homography of a certain instance, mixed point correspondences can be a cause of selecting an incorrect solution in RANSAC. The next section describes RANSAC in which an evaluation on appearances of ads is incorporated to select the correct solution.

2.2 Putative Alignment by RANSAC with Appearance Evaluation

We denote point correspondences as a set using new index k; $C = \{\boldsymbol{p}_k^m, \boldsymbol{p}_k^s\}$, $k = 1, \ldots, P$. Expressing the homography that takes each $\boldsymbol{p}_k^m$ to $\boldsymbol{p}_k^s$ as $\boldsymbol{H}$ (3×3 matrix), transformation error is defined by the following equation:

$$e_k = \|\boldsymbol{p}_k^s - \boldsymbol{H}\boldsymbol{p}_k^m\|\,,\ k = 1, \ldots, P\,. \tag{3}$$



(a) (b)

Fig. 3. Examples of appearances of ads that are rejected in RANSAC. (a) an appearance corresponding to a twisted rectangle, (b) an appearance corresponding to a reversed rectangle. Such appearances are never observed in actual scenes.

We can calculate the position and area of an instance in a scene by transforming a model image by $\boldsymbol{H}$. $\boldsymbol{H}$ can be calculated by the following RANSAC [17]:

(i) A sample comprising of 4 point correspondences is extracted randomly from the set C.

(ii) $\boldsymbol{H}$ is calculated from the sample using Direct Linear Transformation (DLT) algorithm [16] followed by non-linear optimization with the sum of transformation errors defined by Eq. (3) as an evaluation function.

(iii) Transformation errors are calculated for all point correspondences to obtain the number of inliers (votes). Inliers are the point correspondences with the errors that satisfy the following equation:

$$e_k < \varepsilon,\ k = 1, \ldots, P\,, \tag{4}$$

where, ε is the threshold value.

(iv) Processes (i) to (iii) are repeated to obtain the inliers with the maximum vote.

(v) $\boldsymbol{H}$ is calculated using the inliers obtained in (iv).

Since point correspondences of instances are mixed as in Fig. 2(b), a sample containing outliers may have the maximum vote by chance in the above algorithm. It is important to note that, in many cases, the results of transforming a model image by $\boldsymbol{H}$ computed from samples extracted from "multiple instances" yield appearances of ads which are never observed in actual scenes. Figure 3 (a) and (b) show the examples of such appearances. The major cause to take such appearances the maximum vote is that the transformation error of Eq.(3) is only criterion to select $\boldsymbol{H}$. To address the problem, the following process evaluating appearances of ads is added after (ii).

(ii') If the result of transforming a model image by $\boldsymbol{H}$ is a twisted rectangle or a reversed rectangle, the process returns to (i). If not, the process proceeds to (iii).

Twisted and reversed rectangles correspond to appearances such as Fig. 3 (a) and (b), respectively. Thus we can avoid voting by Eq. (4) for homographies corresponding to impossible appearances by the process (ii'). Twisted rectangles can be detected by whether the intersection points of lines obtained by connecting the vertexes of a model image after transformation are within the convex closure comprising of transformed vertexes. Reversed rectangle can be detected by the signed area [20] used to find the faces of polygons. Since the computations to detect these rectangles are very efficient, the evaluation in (ii') is suited to RANSAC which requires iterations.

We calculated homographies from 10000 samples obtained from the point correspondences in Fig. 2 (b), and found 9899 homographies corresponding to twisted or reversed rectangles. Since many impossible appearances actually occur in RANSAC as seen in this case, the evaluation in (ii') is really effective for selecting the correct solution. Figure 2 (c) shows the result of putative alignment in the case of $\varepsilon = 3$ [pixel] in Eq. (4). In this figure, the lines show inliers and the rectangle is the result of transforming the model image by the homography obtained by RANSAC with appearance evaluation. The false matches and mixed point correspondences have been removed and the single instance is separated successfully.

2.3 Guided Matching

Using the result of putative alignment such as Fig. 2 (c), we can obtain point correspondences only from regions around a single instance, which excludes the effects of a background and other instances. For local invariant features of a model image, predicted positions for matching are computed using the homography $\boldsymbol{H}$ obtained in putative alignment as:

$$\hat{\boldsymbol{p}}_k^s = \boldsymbol{H}\boldsymbol{p}_k^m,\ k = 1, \ldots, P\,. \tag{5}$$

We can set circular search regions with the predicted positions as the centers and radii r_k in a scene image. The radii r_k of the search regions are determined by the scale of the features as $20\sigma_k^m$. The distances between descriptors are calculated only for local invariant features in the circular search regions and they are evaluated by Eqs. (1) and (2). If only one point correspondence is found and thus Eq. (1) cannot be evaluated, the point correspondence shall be used.

2.4 Final Alignment and Verification

Using the point correspondences obtained by guided matching, we calculate a homography again by RANSAC with appearance evaluation. Then we verify the alignment result. Circular regions are prepared for N_i inliers by the same way as guided matching. The similarity between a model image transformed by $\boldsymbol{H}$ and a scene image are measured by the normalized cross correlations of RGB channels computed within the regions. Note that the calculation of the similarity shown here can minimize the effects of occlusions, because the normalized cross correlations are calculated in the local regions instead of the entire image. If the following equation on the average value of the normalized cross correlations, $NCC_l, l = 1, \ldots, N_i$, is satisfied, the final alignment result is accepted:

$$\frac{1}{N_i}\sum_{l=1}^{N_i} NCC_l > \gamma\,, \tag{6}$$

where, γ is the threshold value. We set $\gamma = 1$.

Figure 2 (d) shows the results of final alignment. Compared to Fig. 2 (c), a homography is calculated with more inliers which lead to accurate position of the instance. The alignment result is accepted, because the average value of NCC_l in Eq. (6) is 2.3. Using the processes in the Sections 2.1 to 2.4, the single instance has been successfully separated although there are multiple instances in Fig. 2.

2.5 Termination Conditions

If the result of final alignment is accepted, the point correspondences in the area of a recognized instance (e.g., in the rectangle that shows the recognition result in Fig. 2 (d)) are removed. To recognize other instances, the processes in Sections 2.2 to 2.4 are performed for the remaining point correspondences. This procedure is repeated until one of the following termination conditions is satisfied: (a) 4 or more inliers cannot be obtained in RANSAC and (b) the condition of Eq. (6) is not satisfied. These conditions correspond to the case with no point correspondences that satisfy the global constraint and the case with an incorrect alignment result, respectively.

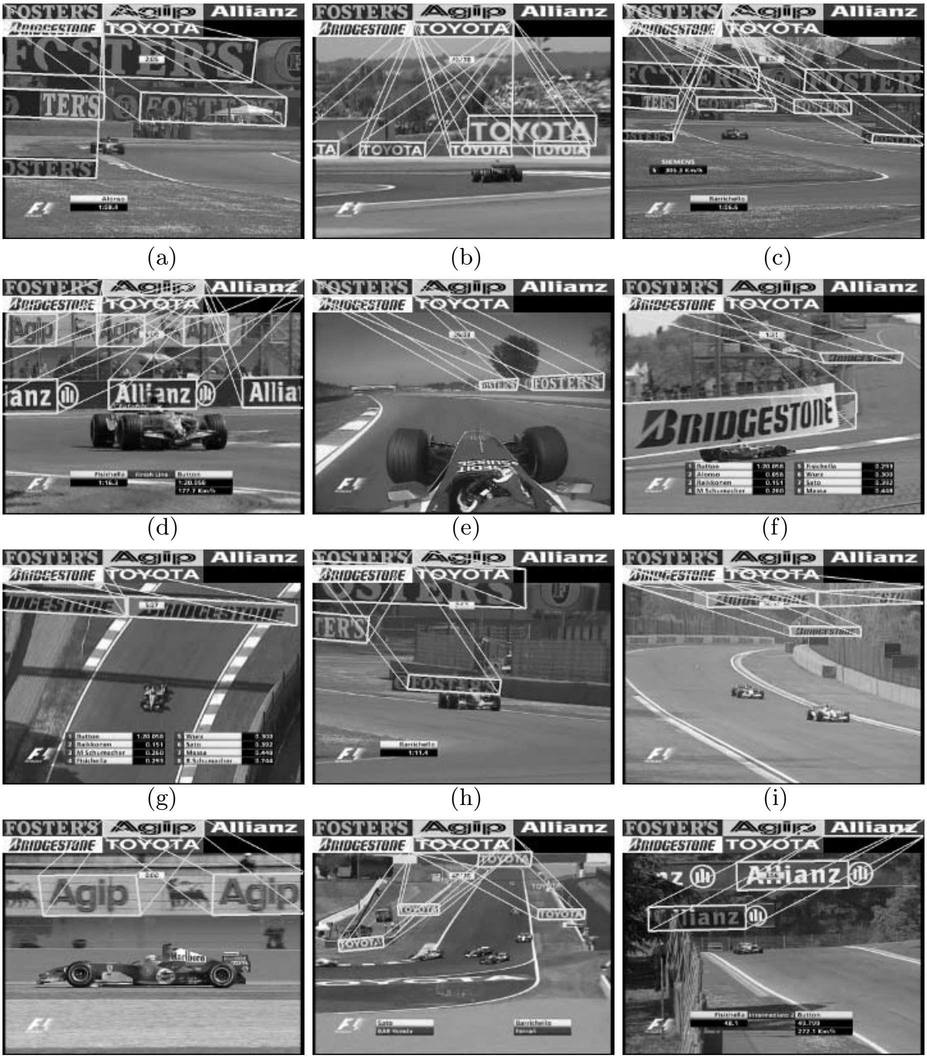
3 Experimental Results

We applied the proposed algorithm to videos of F1. Five ads were selected as recognition targets, and the model images shown at the top of each image in Fig. 4 were used. For each target, only one model image shown in Fig. 4 was given .

Figures 4 (a)∼(j) show the successful results. In spite of deformations, illumination changes and occlusions of the ads in these scene images, all ads were successfully recognized by separating point correspondences of each instance. These results demonstrate that our method recognizes multiple ads in various situations even if only one model image is given.

Figures 4 (k) and (l) show negative examples. Since the view point was located horizontally against the ad on the ground in front in Fig. 4 (k), the deformation of the ad is extremely large. Ads located at a far distance are observed as extremely small in size. The deformation and scaling for these ads seemed to have exceeded the range that could be compensated by the invariant property of the local features, and thus point correspondences were not obtained. In Fig. 4 (l), recognition of the top left ad failed. In this case, the degree of occlusion was too large to obtain the sufficient number of point correspondences.

As seen in the negative examples of Figs. 4 (k) and (l), recognition naturally fails if point correspondences cannot be obtained even by using local invariant features. One method for addressing such situations is to develop a local invariant feature that is better able to deal with deformations and occlusions. Another promising method is to use several model images including the deformations of ads that can be expected beforehand. We are currently working to examine these two methods.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 4. Recognition results for broadcasts of F1. The rectangles in the figures show the recognition results. Figures (a) to (j) show successful cases. In spite of deformations, illumination changes and occlusions of the ads in these scene images, all ads were successfully recognized by separating point correspondences of each instance. These results demonstrate that our method recognizes multiple ads in various situations even if only one model image is given. Figures (k) and (l) show failure cases. (k) Since the ranges of deformation and scaling that could be covered by the invariants were exceeded, point correspondences could not be obtained for the ad on the ground in front and the small ads in distance. (l) The degree of occlusion was large and recognition of the top left ad failed. Such failures may be eliminated by improving the local invariant features and using several model images including the deformations of targets that can be expected beforehand.

4 Summary

In this paper, we have presented an algorithm for recognizing multiple billboard advertisements in videos. We used the local invariant features for matching between a model image and a scene image, but false matches and mixed point correspondences appeared due to the effect of a background and multiple instances of ads. To separate the point correspondences of a single instance from the result of matching, we introduced the outlier rejection method which yields the reasonable number of point correpondences so that we can use RANSAC to calculate a homography. We also introduced RANSAC with appearance evaluation in which impossible appearances of ads are rejected. Final alignment and verification were done using the point correspondences found by guided matching based on the homography. These procedures were carried out sequentially until the termination condition was satisfied. The experimental results showed the usefulness of our algorithm. We believe that the algorithm presented here will be a good tool for several applications such as marketing research and video retrieval.

References

1. C. Harris and G. Giraudon. A combined corner and edge detector. In *Proc. 4th Alvey Vis. Conf.*, pages 147–151, 1988.
2. C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Trans. PAMI*, 19(5):530–535, 1997.
3. D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, volume 2, pages 1150–1157, 1999.
4. M. Brown and D. Lowe. Invariant features from interest point groups. In *Proc. BMVC*, pages 253–262, 2002.
5. J. Sivic and A. Zisserman. Video google: a text retrieval approarch to object matching in videos. In *Proc. ICCV*, volume 2, pages 1470–1477, 2003.
6. M. Brown and D. Lowe. Recognising panoramas. In *Proc. ICCV*, volume 2, pages 1218–1225, 2003.
7. D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
8. K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.
9. M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *Proc. CVPR*, volume 1, pages 510–517, 2005.
10. P. Quelhas, F. Monay, J.M.Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. In *Proc. ICCV*, volume 1, pages 883–890, 2005.
11. F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proc. ICCV*, volume 1, pages 636–643, 2001.
12. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, pages 384–393, 2002.
13. T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *IJCV*, 59(1):61–85, 2004.
14. K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *Proc. BMVC*, volume 2, pages 779–788, 2003.

15. P. H. S. Torr. Geometric motion segmentation and model selection. *Phil. Trans. R. Soc. Lond. A*, 356:1321–1340, 1998.
16. R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition, 2003.
17. M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *ACM Graphics and Image Processing*, 24(6):381–395, 1981.
18. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. ICCV*, volume 1, pages 525–531, 2001.
19. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proc. CVPR*, pages 257–264, 2003.
20. T. Moller and E. Haines. *Real-time rendering*. A.K.Peters, 2nd edition, 2002.

Occlusion Detection and Tracking Method Based on Bayesian Decision Theory*

Yan Zhou, Bo Hu, and Jianqiu Zhang

Department of Electronic Engineering, Fudan University, Shanghai, 200433
Phone: 021-65642762; Fax: 021-65642143
051021012@fudan.edu.cn

Abstract. In order to track an occluded target in an image sequence, the Bayesian decision theory is, here, introduced to the problem of distinguishing occlusions and appearance changes according to their different risk possibilities. A new target template combining image intensity and histogram is designed. The corresponding updating method is also derived based on particle filter. If the target is totally occluded by another target, the template can be kept unchanged. The occlusion of a target will not influence tracking. Simulation results show that the presented method can efficiently justify whether the occlusion occurs and realize target tracking in image sequences even though the tracked target is totally occluded with long time.

1 Introduction

Target tracking is one of the most attractive topics in the computer vision area. The main task is to detect, track, and recognize the target and understand its behavior, which is widely used in transportation surveillance systems, robot visual navigation, safety control, medical diagnosis and weather analysis.

The target tracking procedure is generally in two steps: template matching and template updating. The different selections of template and template updating methods yield to different performances. There are several kinds of templates existed based on image features, such as contour based template [1], color based template, motion based template [5], and mixed features based template(Wolfe[2] biologic vision model).

In target tracking field, mathematic models can be divided into two groups: deterministic tracking and stochastic tracking. Deterministic approaches usually

* The research is sponsored by 973 National Basic Research Program. Program No. 2006CB705700. Project name: Research on Key Scientific and technologic Problems of molecular imaging.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 474 – 482, 2006.

reduce to an optimization problem, e.g., minimizing an appropriate cost function [3]. Mean shift [6] is an alternative deterministic approach to visual tracking, where the cost function is derived from the color histogram. Stochastic tracking approaches often reduce to an estimation problem, e.g., estimating the state for a time series state space model [3]. For linear systems with Gaussian noises, Kalman Filter could find optimal solution. But in many situations of interest, the assumptions made above do not hold. Therefore, approximations are necessary. Extended Kalman Filter was proposed which can obtain suboptimal solution. Sequential Monte Carlo method was widely used in nonlinear/non-Gaussian systems for its flexibility and simplicity, namely Particle Filter [4]. The key idea of Particle Filter is to represent the required posterior density function by a set of random samples and associated weights.

However, Particle Filter falls apart under severe occlusions. Occlusion problem is hard to handle in tracking because both occlusions and target appearance changes can cause intensity changes in image sequences. The occlusion detection method is too simple in [3] so that it gradually loses the track of the target to the occluding object.

In this paper, a histogram and intensity mixed template is proposed to combine the good qualities of both intensity and histogram. Intensity is easy to obtain, and fine enough to describe the object appearance, whereas histogram is a statistical description of the object, which is flexible to partial intensity change and scale variety. Combing the two features makes tracking more stable. If an occlusion is declared, the template will not update until the object reappear.

Based on this template, the Bayesian decision theory is introduced to distinguishing occlusions and appearance changes according to their different risk possibilities. That is, mistaking occlusions for appearance changes will cause the template to update, thus with a high risk to lose the target. But mistaking appearance changes for occlusions will only cause the template unchanged, which has little influence in the tracking procedure, thus with a comparatively low risk. In this way, occlusions can be efficiently detected even if some appearance changes would be judged as occlusions.

To calculate the occlusion probability, we adopt a block matching method based on the fact that appearance changes happens randomly in the whole target area, but occlusion always happens from one side of the target.

This paper is organized as follows. We introduce system framework in Section 2. We focus on the solution of occlusion detection problem in Section 3. Experimental results will be shown in Section 4, with conclusions presented in Section 5.

2 System Framework

The framework of proposed algorithm shows in Fig1, which mainly involves template matching, occlusion detection followed by state estimation procedure.

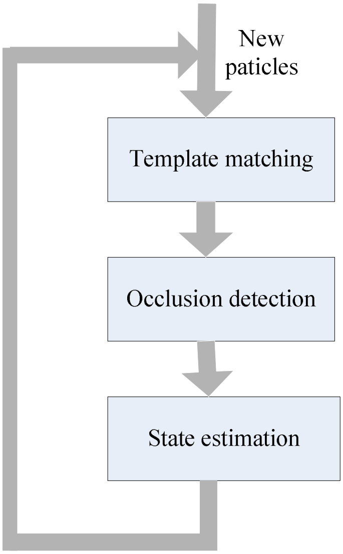


Fig. 1. The algorithm framework. It works in three steps, template matching, occlusion detection and state estimation.

2.1 Particle Filter

Define x_k , z_k as the target state and target observation at time k . In the target tracking problem, we are interested in the posterior $p(x_k \mid z_{1:k})$, where $z_{1:k} = (z_1 \cdots z_k)$ are observations up to time k . The key idea of Particle Filter is to represent the required posterior density function $p(x_{k-1} \mid z_{1:k-1})$ by a set of random samples and associated weights $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N$. New samples are generated according to an importance function, which relies on the state transition function and observations $x_k^i \sim q(x_k \mid x_{k-1}^i, z_k), i = 1 \ldots N$. The weights of new samples are updated using the function below:

$$w_k^i \propto w_{k-1}^i \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} \quad (1)$$

It produces a new set of weighted samples as an approximation to $p(x_k \mid z_{1:k})$. In this paper, we choose the importance function to be $q(x_k^i \mid x_{k-1}^i, z_k) = p(x_k^i \mid x_{k-1}^i)$, namely, Bootstrap Particle Filter. Detailed explanations for Bootstrap Particle Filter are available in [4]. The main procedure is summarized here:

```
The samples set $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N$ and estimated state at time
$k-1$ are known.
For $i=1\ldots N$
Generate new samples: $x_k^i = \tilde{x}_{k-1} + v_k^i$;      ( $v_k^i$ is Gaussian noise)
Update sample weights:   $w_k^i \propto w_{k-1}^i p(z_k \mid x_k^i)$;  ( $\sum_{i=1}^N w_k^i = 1$ )
End
Estimate the target state $\tilde{x}_k$ at time $k$;
```

In the paper, we use the state transition model (2) to generate new samples at each time step.

$$x_k^i = \tilde{x}_{k-1} + v_k^i \tag{2}$$

Where v_k^i follows Gaussian distribution. Template matching coefficient is used as sample weight, which will be mentioned in detail in Section 2.2. We implement MAP rule to estimate the target state.

$$\tilde{x}_k = x_k^{map} = \arg\max_{x_k} p(x_k \mid z_{1:k}) \approx \arg\max_{x_k} w_k^i \tag{3}$$

2.2 Histogram and Intensity Mixed Template

The task is to track a specific target no matter they are rigid such as vehicles or non-rigid such as faces and people. So our approach reaches a wide range of application situations. A histogram and intensity mixed template is proposed, here, to combine the good qualities of both intensity and histogram. Intensity is easy to obtain, and fine enough to describe the object appearance, whereas histogram is a statistical description of the object, which is flexible to partial intensity change and scale variety. After a set of samples are generated at time k, a mixed correlation coefficient is computed using (4).

$$\lambda_k^i = \beta \cdot r_t + (1-\beta) \cdot r_h \tag{4}$$

Where r_t represents the correlation coefficient of intensity, and r_h represents the correlation coefficient of histogram. The choice of the parameter β is done in an empirical way (β equal to 0.8).

3 Occlusion Detection

In a tracking procedure, a target is always occluded by other objects, which will probably result in the lost of the target. So it is important to detect occlusion. Occlusion problem is hard to handle in tracking because both occlusions and target appearance changes can cause intensity changes in image sequences. Bayesian decision theory is, here, introduced to the problem of distinguishing occlusions and appearance changes according to their different risk possibilities. In this framework, a block matching method is proposed to compute the occlusion probability. A correspondingly template updating rule is also used in our approach.

3.1 Bayesian Decision Theory

In a tracking problem, occlusions and appearance changes are taken as two classes. Then, a classifier is designed to distinguish them. Instead of minimizing total classification errors, Bayesian decision theory is a rule which aims at minimizing total classification risk. Our approach uses Bayesian decision theory as the decision rule on account of the fact that mistaking occlusions for appearance changes will cause the template to update, thus with a high risk to lose the target. But mistaking appearance changes for occlusions will only cause the template unchanged, which has little influence in the tracking procedure, thus with a comparatively low risk.

Let $\{c_1, c_2\}$ be occlusions and appearance changes. $\lambda(c_1 c_2)$ defines the risk coefficient while mistaking occlusions for appearance changes. $\lambda(c_2 c_1)$ defines the risk coefficient while mistaking appearance changes for occlusions. Then the discriminate function is as follows:

$$g(x) = p(c_1 \mid x)\lambda(c_2 c_1) - p(c_2 \mid x)\lambda(c_1 c_2) \tag{5}$$

When $g(x) > 0$, classify to c_2 , when $g(x) < 0$, classify to c_1 . $p(c_1 \mid x)$ is the occlusion probability, and $p(c_2 \mid x)$ indicates appearance change probability. As we illustrated before, $\lambda(c_1c_2)$ is higher than $\lambda(c_2c_1)$. In the experiment, $\lambda(c_1c_2)$ is set to 5 and $\lambda(c_2c_1)$ set to 1.

3.2 Block Matching Method

To calculate the occlusion probability $p(c_1 \mid x)$, we adopt a block matching method based on the fact that appearance changes pixel intensity randomly in the whole target area, but occlusion always happens from one side(left, right, up or down) of the target, so pixel intensity at one side changes much more than the inner area. Therefore, we divide the target into four blocks(Fig2); the occlusion probability function is defined:

$$p(c_1 \mid x) = \min(\frac{\max(\mu) - \min(\mu)}{\alpha}, 0.99) \tag{6}$$

Where μ is a vector containing four block matching coefficients. α is a constant to make sure $\frac{\max(\mu) - \min(\mu)}{\alpha}$ is in (0,1). If one side of the target is occluded, $\max(\mu) - \min(\mu)$ becomes larger indicating a high probability of occlusions. if four coefficients drop simultaneously, $\max(\mu) - \min(\mu)$ changes little because it is more likely to be caused by appearance changes, not occlusions. After $p(c_1 \mid x)$ is computed, $p(c_2 \mid x) = 1 - p(c_1 \mid x)$.

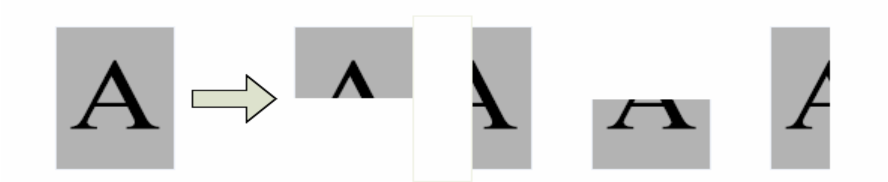


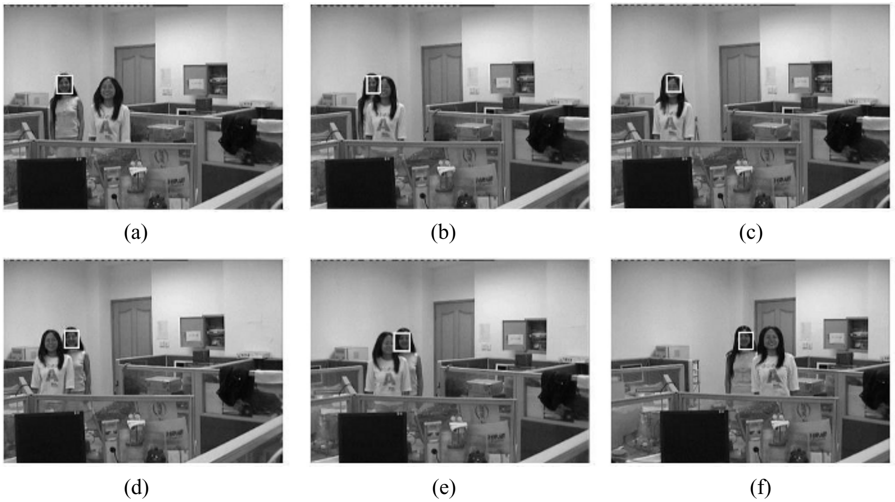
Fig. 2. Block matching method. The four blocks represents the fact that an occlusion always happens from one side of the target, left, right, up or down.

3.3 Template Updating Rule

The idea above yields the corresponding template updating rule: if the matching coefficient is higher than the threshold T_0, the template will update. If not, using the Bayesian decision theory to detect occlusions. When an occlusion is declared, the template and the target state will stay unchanged until the coefficient becomes higher than the threshold T_1 , which is smaller than T_0 , we empirically define it to be $T_1 = 0.7T_0$. T_0 is obtained by averaging the initial 5 frames of the image sequences. It is reasonable to suppose there is no occlusion in these 5 frames.

4 Experimental Results

This section presents the experimental results of the tracking procedure. In our implementation, we use the following choices. The state space is $x = (x^*, y^*, \alpha^*)$, where x^* and y^* are 2-D translation parameters, α^* is the scale parameter. Their initial values are set manually in the first frame. We implemented our approach in many image sequences. Fig3 shows the result of a face tracking. During the tracking, one face is totally occluded by another face. Fig4 shows the result of a person tracking at night in outdoor environment, during which the person is totally occluded by a head for a few frames.



(a) (b) (c)

(d) (e) (f)

Fig. 3. (a) is the first frame of the tracking. (b) is the20th frame, an occlusion is declared, then the template and the target state stay unchanged. (c) is the 27th frame, the face is totally occluded by another face. (d) is the 40th frame, after the target reappears, the tracking goes on. (e) is the 58th frame, another occlusion happens. (f) is the 81st frame.

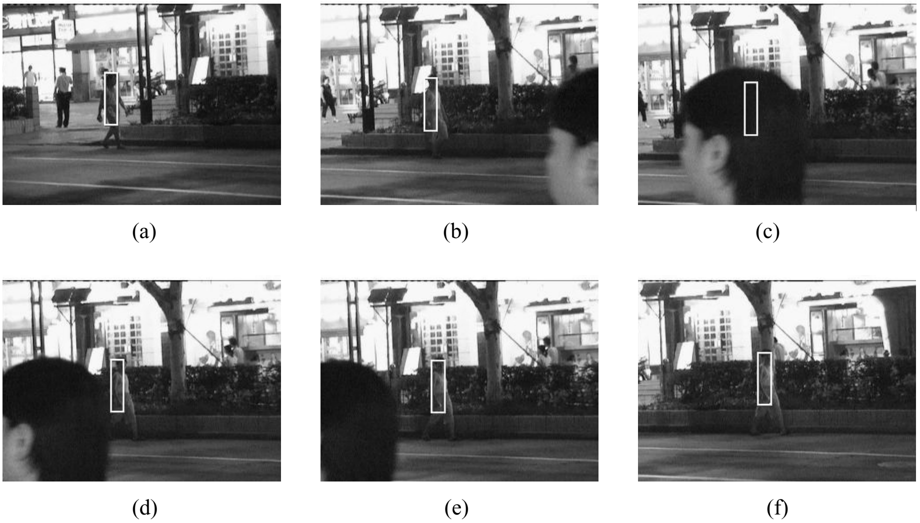


Fig. 4. (a) is the 200th frame of the tracking. (b) is the262th frame (c) is the 268th frame, total occlusion happens. Because the template and the target state stay unchanged during occlusion, the state variable in the 270th frame(d) is the same as that in (c). Then in the 271st frame(e), reappearance is declared, so the state variable begins to update. (f) is the 298th frame.

5 Conclusion

We have implemented Bayesian Theory in the occlusion detection problem, and a block matching method is proposed to compute the occlusion probability. A target template combining image intensity and histogram and the corresponding template updating method are used in this paper. When the presented template and the corresponding template updating rules are exploited in the particle filter-based tracking algorithm, the occlusion of a target will not influence it tracked. The experimental results show that our method can efficiently justify whether the occlusion occurs and realize target tracking in image sequences even though the tracked target is totally occluded with long time.

References

1. Michael Acheson Isard Robotics Research Group: Visual Motion Analysis by Probabilistic Propagation of Conditional Density. PhD thesis.1998
2. Wolfe, J. M. (1994), ‘Guided Search 2.0: A revised model of visual search’, Psychonomic Bulletin & Review 1(2), 202–238
3. Shaohua Zhou, Rama Chellappa, Baback Moghaddam: Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters. IEEE Transactions on Image Processing, 13:11,pps. 1491-1506, 2004

4. M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp: A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Transactions on Signal Processing, Vol.50, NO.2, February 2002
5. Weiming Hu, Tieniu Tan: A Survey on Visual Surveillance of Object Motion and Behaviors. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and reviews, Vol.34, NO.3, August 2004
6. Yizong Cheng: Mean Shift, Mode Seeking, and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, NO. 8, August 1995
7. ARNAUD DOUCET, SIMON GODSIL and CHRISTOPHE ANDRIEU: On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. Statistics and Computing (2000) 10, 197–208.
8. Morelande, M.R.; Challa, S.: Manoeuvring target tracking in clutter using particle filters. IEEE Transactions on Aerospace and Electronic Systems Volume 41, Issue 1, Jan. 2005 Page(s):252 - 270

Pedestrian Recognition in Far-Infrared Images by Combining Boosting-Based Detection and Skeleton-Based Stochastic Tracking

Ryusuke Miyamoto[1,2], Hiroki Sugano[1,2], Hiroaki Saito[3], Hiroshi Tsutsui[1],
Hiroyuki Ochi[1,2], Ken'ichi Hatanaka[3], and Yukihiro Nakamura[1,2]

[1] Dept. of Communications and Computer Engineering, Kyoto University,
Yoshida-hon-machi, Sakyo, Kyoto, 606-8501, Japan
{miya, hiroki, tsutsui}@easter.kuee.kyoto-u.ac.jp,
{ochi, nakamura}@kuee.kyoto-u.ac.jp
[2] Kyoto Center, Synthesis Corporation,
22-75-201, Tanakasekitachou, Sakyo, Kyoto, 606-8203, Japan
{miya, hiroki, ochi, nakamura}@synthesis.co.jp
[3] Sumitomo Electric Industries, Ltd.
1-1-3, Shimaya, Konohana-ku, Osaka, 554-0024, Japan
{saito-hiroaki, khata}@sei.co.jp

Abstract. Nowadays, pedestrian recognition in far-infrared images toward realizing a night vision system becomes a hot topic. However, sufficient performance could not be achieved by conventional schemes for pedestrian recognition in far-infrared images. Since the properties of far-infrared images are different from visible images, it is not known what kind of scheme is suitable for pedestrian recognition in far-infrared images. In this paper, a novel pedestrian recognition scheme combining boosting-based detection and skeleton-based stochastic tracking suitable for recognition in far-infrared images is proposed. Experimental results by using far-infrared sequences show the proposed scheme achieves highly accurate pedestrian recognition by combining accurate detection with few false positives and accurate tracking.

1 Introduction

Nowadays, many kinds of night vision systems to assist drivers at nighttime are developed[1,2,3,4,5,6,7], and using a far-infrared camera in such systems becomes popular. However, performance of conventional methods is insufficient for night vision systems since pedestrian recognition problem itself is difficult and the properties of images acquired from a far-infrared camera are different from visible images. Therefore, a highly accurate scheme suitable for pedestrian recognition in far-infrared images is required.

Pedestrian recognition based on image processing is widely tackled [1,2,3,4,5,6, 7,8,9,10,11,12]. [8,9,10,11,12], [6], and [1,2,3,4,5,7] aim pedestrian recognition in visible, near-infrared, and far-infrared images, respectively. Generally, pedestrian recognition is performed by three steps as same as object recognition in visual

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 483–494, 2006.

surveillance[13]; candidate extraction, pedestrian detection from candidate areas, and tracking of detected pedestrians. Candidate extraction enhances the accuracy of pedestrian detection since areas which obviously include no pedestrian can be ignored before detection process and tracking enhances total performance of pedestrian recognition since a pedestrian who is once detected can be recognized even if it is not detected successfully in some subsequent frames.

Stereo vision is widely used in candidate extraction[2,4,10,11]. However, a dense disparity map cannot be obtained by stereo matching in far-infrared images because of the properties of themselves. Therefore, it is difficult to extract candidate regions by stereo segmentation in far-infrared images. In [1], which adopts a monocular far-infrared camera, candidate regions called hotspots are extracted by binarization based on the fact that luminance of the head of a pedestrian is high. This scheme is simple but sometimes does not work successfully.

For detection phase, neural network is adopted in [11], Support Vector Machine (SVM) is applied in [8,9,10,1], and boosting-based scheme are proposed in [12]. Most of these schemes aim pedestrian detection in visible images and the scheme suitable for pedestrian detection in far-infrared images has not been proposed.

For tracking phase, Kalman filter is widely adopted [14,1,10]. However, due to the fact that functional assumptions of models used in tracking based on Kalman filter (linearity, Gaussianity, and unimodality) are often violated in pedestrian tracking, Kalman filter can not achieve accurate pedestrian tracking in principle. Meanwhile, in the field of object tracking[15,16,17,18], particle filter[19] has been attempted recently. Since the assumptions tracking based on Kalman filter depends on are no more required, this approach is expected to solve problems, which existing tracking schemes have been confronted with, such as occlusions and non-linear motion of a tracking target. Therefore, to enhance the accuracy, a tracking scheme based on particle filter suitable for pedestrian tracking in far-infrared images is required.

In this paper, a novel pedestrian recognition scheme combining accurate pedestrian classifier with which pedestrians are detected by searching whole of each input image without candidate extraction and tracking based on particle filter which is expected to achieve highly accurate tracking is proposed. The accurate classifier required for detection with searching whole of each input image is constructed by boosting in the proposed scheme. For the tracking phase, skeleton-based stochastic tracking which adopts particle filter[20] for state estimation proposed by the authors is applied. The proposed scheme is evaluated by applying it to sequences acquired by a far-infrared camera.

The rest of this paper is organized as follows. Section 2 describes boosting and skeleton-based stochastic tracking which adopts particle filter. In Section 3, the boosting-based pedestrian detection used in the proposed scheme is described and the performance of pedestrian detection is evaluated by applying it to far-infrared sequences. In section 4, the tracking accuracy of skeleton-based scheme in far-infrared images is evaluated and a skeleton model suitable for the tracking is selected. In section 5, the boosting-based detection and the skeleton-based tracking are combined with automatic initialization required for the skeleton-based tracking and it is shown that the proposed scheme detects and tracks pedestrians properly. This paper is concluded with Section 6.

2 Preliminaries

In the proposed recognition scheme, boosting-based detection and skeleton-based stochastic tracking are adopted, since boosting is one of the most successful classification methods used in pattern recognition and the skeleton-based tracking achieves highly accurate pedestrian tracking in visible images. Especially, since the skeleton-based tracking requires only input binary images which represent silhouettes of tracking targets, it is expected to work properly in far-infrared images. In this section, overview of boosting and skeleton-based tracking used in the proposed scheme is described.

2.1 Boosting

Boosting is one of the ensemble learning methods with which accurate classifier is constructed by combining weak hypotheses learned by weak learning algorithm. Obtained classifier consists of weak hypotheses and a combiner, and output is computed by weighted vote of weak hypotheses. In the proposed scheme, AdaBoost[21], one of the most popular methods based on boosting, is adopted for construction of an accurate classifier. The learning flow of AdaBoost is shown as follows.

Algorithm 2.1. AdaBoost$(h, H, (x_1, y_1), \ldots, (x_n, y_n), m, l, T)$

```
for $i \leftarrow 1$ to $n$
  do if $y_i == 1$
  then $w_{1,i} = \frac{1}{2m}$
  else $w_{1,i} = \frac{1}{2l}$
for $t \leftarrow 1$ to $T$
  do { for $i \leftarrow 1$ to $n$
         do $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
       for $j \leftarrow 1$ to $H$
         do $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$
       Choose the classifier $h_t$, with the lowest error $\epsilon_t$
       for $i \leftarrow 1$ to $n$
         do $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}, \beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
       where $e_i = 0$ if example $x_i$ is classified correctly, $e_i = 1$ otherwise
Final strong classifier is: $h(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t, \alpha_t = \log 1/\beta_t \\ 0 & \text{otherwise} \end{cases}$
```

where x is an input sample and y indicates a label of the sample. Input is negative sample if $y = 0$, and input is positive sample if $y = 1$. T is the number of iteration, m and l are the number of negative and positive examples, respectively, h is a set of weak classifier, and H is the number of sets of weak classifier.

2.2 Skeleton-Based Tracking

A definition of state space, a rule of state transition, a scheme of likelihood estimation must be provided to realize tracking based on particle filter. In [20], a skeleton model shown in Fig. 1 is adopted and state space $\boldsymbol{X}$ is defined by the following:

$$\boldsymbol{X} \overset{\triangle}{=} \{x_{\mathrm{B}}, y_{\mathrm{B}}, l_{\mathrm{AB}}, l_{\mathrm{BC}}, l_{\mathrm{BE}}, l_{\mathrm{EF}}, a_{\mathrm{AB}}, a_{\mathrm{BC}}, a_{\mathrm{BD}}, a_{\mathrm{BE}}, a_{\mathrm{EF}}, a_{\mathrm{EG}}\},$$

where an edge between nodes P and Q is denoted as stick$_{\mathrm{PQ}}$, while l_{PQ} and a_{PQ} indicate the length of stick$_{\mathrm{PQ}}$ and the angle of the vector $\overrightarrow{\mathrm{PQ}}$ of stick$_{\mathrm{PQ}}$ to x axis, respectively. $(x_{\mathrm{B}}, y_{\mathrm{B}})$ denotes the absolute coordinate of the node B.

For the state transitions of x_{B}, y_{B}, and a_{PQ}, 2nd-order autoregressive model is used with the following restrictions:

$$\pi/4 \leq a_{\mathrm{AB}}, a_{\mathrm{BE}} \leq 3\pi/4$$
$$0 \leq a_{\mathrm{BC}}, a_{\mathrm{BD}}, a_{\mathrm{EF}}, a_{\mathrm{EG}} \leq \pi.$$

1st-order autoregressive model is applied for the state transitions of l_{PQ}.

In likelihood estimation, first, a silhouette of a tracking target is extracted as a binary image and a distance transformed image is generated. An example of a binary and a distance transformed image are shown in Fig. 2 (a) and Fig. 2 (b), respectively. Next, several points on each predicted stick are selected, and a point which has the maximum pixel value on the orthogonal direction to a stick in the distance transformed image is searched for each point as illustrated in Fig. 2 (c). Then, $e^{(n)}$ is calculated by the following:

$$e^{(n)} = \exp\Big\{-\frac{1}{r}\sum_i \big(\min(d_i^{(n)}, \mu)\big)^2\Big\},$$

where d_i denotes the distance between the selected point and the searched point, i corresponds to an index of selected points, r represents a constant value for scaling parameter, n indicates the number of particles, and μ is a search range. Finally, a likelihood $w^{(n)}$ of each particle is obtained by normalizing $e^{(n)}$ by $w^{(n)} = e^{(n)} / \sum_{k=1}^{N} e^{(k)}$.

3 Pedestrian Detection

In order to enhance the performance of detection, both candidate extraction and detection itself must be improved. In this paper, accurate detection without candidate extraction is adopted since to improve candidate extraction is difficult. The proposed method aims to detect pedestrians by searching whole of each input image with an accurate classifier. However, pedestrians can not be detected properly by searching whole of each input image with a SVM-based classifier as shown in Figs. 3 and 4, where rectangles indicate detected pedestrians. Figs. 3 and 4 show the result of SVM-based classification using a gray value and Haar

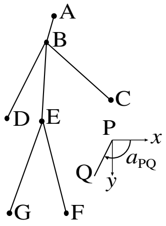
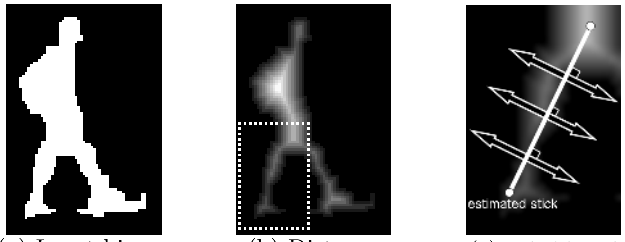


Fig. 1. 6-stick skeleton model



(a) Input binary image

(b) Distance transformed image

(c) Likelihood estimation

Fig. 2. Distance transformation and likelihood estimation

wavelet as a feature vector. To obtain a more accurate classifier, boosting is adopted. In the rest of this section, construction of a classifier, the pedestrian detection by searching whole of each input image with the constructed classifier by boosting, and performance evaluation of pedestrian detection in far-infrared image is described.

3.1 Construction of Classifier

Classifier is constructed by using AdaBoost shown in Section 2.1. In this process, the performance of classification depends on given training samples and features used in learning. In [1], to build multiple classifiers is attempted by classifying training set into three types of pedestrians; along-street pedestrian, across-street pedestrian, and bicyclist. However, there is no significant difference of detection performance between a single classifier and combination of multiple classifiers. Therefore, multiple classifiers are not adopted in this paper. Figs. 5 and 6 show a part of training samples and features adopted for AdaBoost in the proposed scheme, respectively.

3.2 Pedestrian Detection Scheme

In the proposed scheme, pedestrians are detected by searching whole of each input image with a classifier constructed by boosting. This detection is operated by the following:

1. set initial coordinates and a size of detection window,
2. read an image specified with the coordinates and the size,
3. detect pedestrians by applying the classifier with scaled features,
4. move the coordinates,
5. if all coordinates is not searched, go to 2,
6. change the size,
7. if the size is not maximum, go to 2, else search is terminated.

In this paper, 10 × 24 and 89 × 213 are adopted as the minimum size and the maximum size of detection window, respectively.

3.3 Experiment and Evaluation

In this experiment, two test sequences obtained by a far-infrared camera are used. Figs. 7 and 8 show the test sequences, which are called $S1$ and $S2$ in this paper, respectively. The image size of these sequences is 320 × 240. The number of frames of $S1$ and $S2$ is 71 and 107, respectively, and $S1$ includes 65 pedestrian regions and $S2$ includes 202 pedestrian regions. Table 1 shows the number of correct detections of pedestrians (True Positive: TP) and the number of false detections of pedestrians (False Positive: FP) obtained by applying the proposed pedestrian detection to these sequences. The number of TPs and FPs by detection of [1] without tracking is also shown in Table 1 as reference.

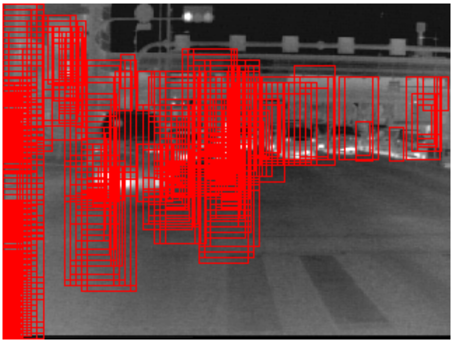


Fig. 3. SVM-based detection by a gray value

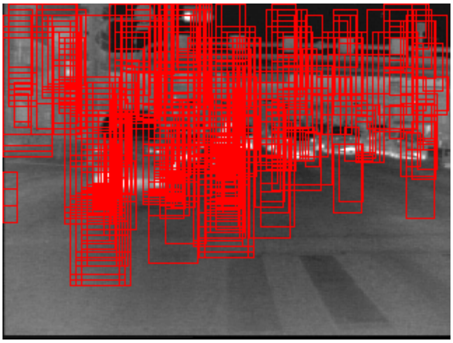


Fig. 4. SVM-based detection by Haar wavelet

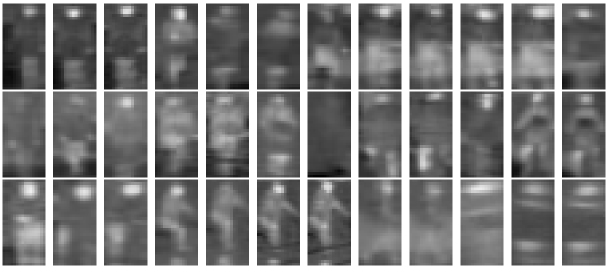


Fig. 5. Training samples

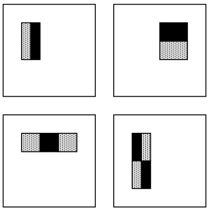


Fig. 6. Features

According to this result, the proposed detection is more accurate than [1] obviously for $S1$. For $S2$, the number of TP by [1] is greater than the proposed detection. However, the number of FP by [1] is much greater than the proposed detection. Considering that FP decreases the accuracy of recognition if tracking is performed consecutively, the proposed detection scheme is superior to the detection scheme of [1] for a pedestrian recognition system which adopts tracking

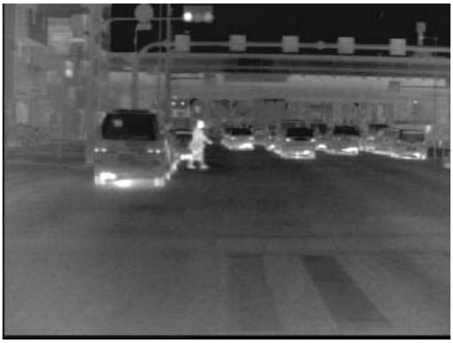


Fig. 7. Sequence $S1$

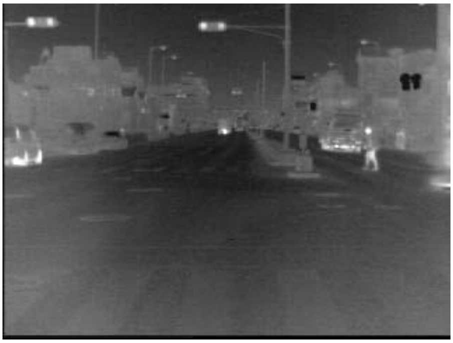


Fig. 8. Sequence $S2$

Table 1. Result of pedestrian detection

scheme	[1]		proposed	
True/False	TP	FP	TP	FP
$S1$	5	6	14	1
$S2$	117	100	69	4

to enhance total performance of recognition. Furthermore, this result shows that classifier constructed by AdaBoost can properly detect pedestrians in far-infrared images with few FPs, though it is shown that such a detection scheme does not work properly for pedestrian detection in visible images in [12].

4 Pedestrian Tracking

A method to extract a silhouette must be defined to perform pedestrian tracking by the skeleton-based tracking since it is required for likelihood calculation. Also, a skeleton model used in tracking suitable for far-infrared images must be determined since the tracking performance depends on an adopted skeleton model[22]. In the rest of this section, how to extract a silhouette and a skeleton model used in this experiment are described and tracking performance is evaluated by using far-infrared images. In this experiment, a state is initialized manually and random numbers required for tracking based on particle filter are generated by Mersenne Twister[23] which is suitable for Monte-Carlo applications.

4.1 Silhouette Extraction and Selection of Skeleton Model

A silhouette extraction scheme by binarization with adaptive threshold for far-infrared images is proposed in [5]. However, this scheme does not work properly under unideal condition. In this paper, simply, interframe subtraction is adopted for silhouette extraction.

Next, in order to select a skeleton model suitable for tracking in far-infrared images, tracking performance by using two types of skeleton model is evaluated. Figs. 13 and 14 show the skeleton model used in this experiment, which are called $M1$ and $M2$ in this paper, respectively.

4.2 Experiment and Evaluation

Figs. 9 and 11 show the tracking result applied to a pedestrian in $S1$ by using $M1$ and $M2$, respectively. Figs. 10 and 12 show the tracking result applied to a pedestrian in $S2$ by using $M1$ and $M2$, respectively. Averages of tracking errors are shown in Table 2.

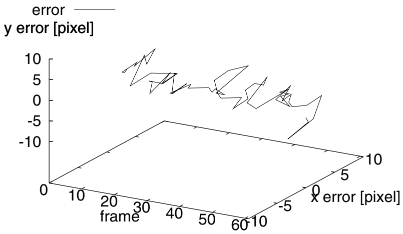


Fig. 9. Tracking result of $S1$ by $M1$

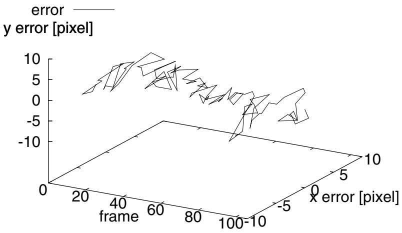


Fig. 10. Tracking result of $S2$ by $M1$

Table 2. Averages of tracking error

model	$M1$		$M2$	
direction	x	y	x	y
$S1$	2.23	2.16	3.49	2.23
$S2$	1.42	1.96	1.31	1.61

This result shows that the tracking is performed properly with low error, and the difference of the tracking accuracy between by $M1$ and by $M2$ is trivial. Since the computational cost required for using $M2$ is less than $M1$, $M2$ is adopted in this paper.

5 Combining Detection and Tracking

In the proposed scheme, pedestrian recognition is achieved by combining detection with few FPs and accurate tracking. The previous sections show that detection with few FPs is achieved by boosting-based detection and accurate tracking is achieved by skeleton-based stochastic tracking. Remaining problem to construct a pedestrian recognition system is to combine boosting-based detection and skeleton-based stochastic tracking. The skeleton-based stochastic tracking requires initialization of a state to start tracking automatically after detection. In the rest of this section, a scheme of initialization is defined and

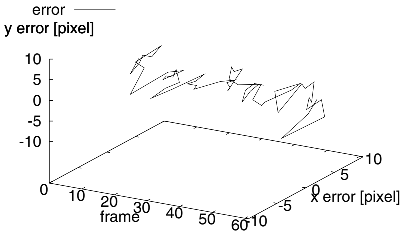


Fig. 11. Tracking result of $S1$ by $M2$

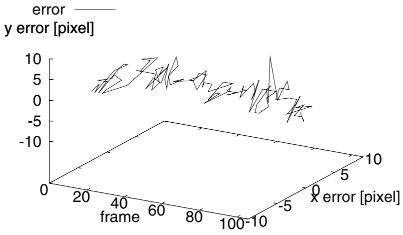


Fig. 12. Tracking result of $S2$ by $M2$

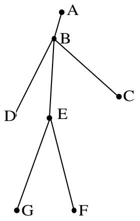


Fig. 13. Skeleton model $M1$

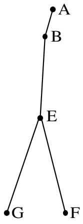


Fig. 14. Skeleton model $M2$

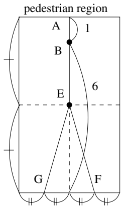


Fig. 15. Initial state of skeleton

then tracking accuracy with this initialization is evaluated by applying to far-infrared sequences.

5.1 Initial Value Setting of Tracking

Since to determine the optimal initial values required for [20] is difficult, in this paper, initial values corresponding to the position of the skeleton as shown in Fig. 15 is adopted, where the rectangle represents a region of detected pedestrian. The initial values given by this scheme is not optimal since the pose of a target pedestrian is unpredictable. However, the number of states corresponding to low likelihood decreases and states corresponding to high likelihood become dominant in tracking process based on particle filter. Therefore it is expected that a target pedestrian is adequately tracked after a while.

5.2 Experiment and Evaluation

Figs. 16 and 17 show the tracking result with the proposed initialization after detection described in the previous section. In this experiment, the same pedestrians as tracked in Section 4 are used. Averages of x and y directional tracking error in $S1$ are 3.20 and 1.59, respectively, and averages of x and y directional tracking error in $S2$ are 1.06 and 1.34, respectively.

By comparison between this result and the result of Section 4, the proposed scheme which tracks a target with automatic initialization after pedestrian detection achieves as same accuracy as the tracking with manual initialization. This

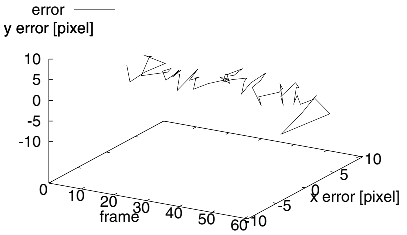


Fig. 16. Tracking result of $S1$ with automatic initialization

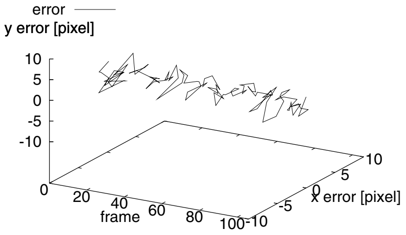


Fig. 17. Tracking result of $S2$ with automatic initialization

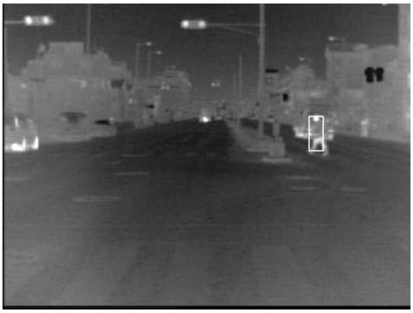


Fig. 18. Recognition result (frame 5)

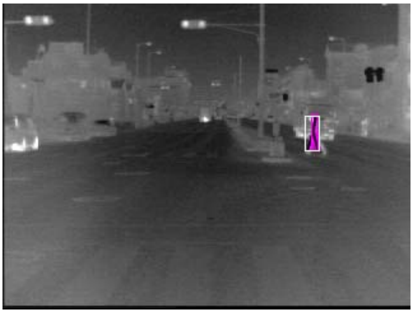


Fig. 19. Recognition result (frame 6)

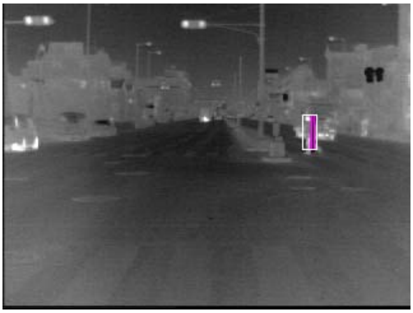


Fig. 20. Recognition result (frame 7)

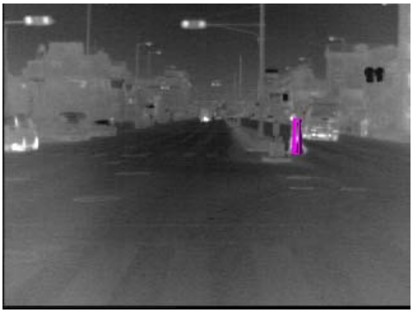


Fig. 21. Recognition result (frame 11)

result shows that the proposed pedestrian recognition scheme which combines boosting-based pedestrian detection and skeleton-based stochastic tracking with automatic initialization works properly in far-infrared images.

Finally, the recognition results by applying the proposed scheme to $S2$ are shown in Figs. 18, 19, 20, and 21. In these figures, rectangles and skeletons show detection results and tracking results, respectively. A pedestrian is detected successfully in Fig. 18, in Figs. 19 and 20, both detection and tracking of the pedestrian succeed, and in Fig. 21, the pedestrian is not detected successfully but can be recognized as pedestrian because he is tracked adequately.

6 Conclusion

In this paper, it is shown that a classifier constructed by AdaBoost detects pedestrians in far-infrared images properly with few FPs compared with the conventional pedestrian detection scheme and the skeleton based tracking proposed in [20] tracks a pedestrian properly with very low error in far-infrared images. The pedestrian recognition scheme combining the boosting-based detection and the skeleton-based tracking with automatic initialization is proposed. Experimental results show that this scheme can track a pedestrian accurately as same as the tracking with manual initialization. These results show that the proposed pedestrian recognition scheme which combines boosting-based pedestrian detection and skeleton-based stochastic tracking works properly in far-infrared images.

Acknowledgements

This work was partly supported by the Japan Society for the Promotion of Science (JSPS) the 21st Century COE Program (Grant No. 14213201).

References

1. Xu, F., Fujimura, K.: Pedestrian detection and tracking with night vision. IEEE Trans. on ITS **6** (2005) 63–71
2. T, T., Hattori, H., Watanabe, M., Nagaoka, N.: Development of night-vision system. IEEE Trans. on ITS **3** (2002) 203–209
3. Fang, Y., Yamada, K., Ninomiya, Y., Horn, B.K.P., Masaki, I.: A shape-independent method for pedestrian detection with far-infrared images. IEEE Trans. on VT **53** (2004) 1679–1697
4. Liu, X., Fujimura, K.: Pedestrian detection using stereo night vision. IEEE Trans. on VT **53** (2004) 1657–1665
5. Yasuno, M., Yasuda, N., Aoki, M.: Pedestrian detection and tracking in far infrared images. In: Proc. of CVPRW. (2004) 125–125
6. Sun, H., Hua, C., Luo, Y.: A multi-stage classifier based algorithm of pedestrian detection in night with a near infrared camera in a moving car. In: Proc. of ICIG. (2004) 120–123
7. Bertozzi, M., Broggi, A., Fascioli, A., Graf, T., Meinecke, M.M.: Pedestrian detection for driver assistance using multiresolution infrared vision. IEEE Trans. on VT **53** (2004) 1666–1678
8. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Proggio, T.: Pedestrian detection using wavelet templates. In: Proc. of CVPR. (1997) 193–199
9. Papageorgiou, C., Poggio, T.: Trainable pedestrian detection. In: Proc. of ICIP. Volume 4. (1999) 35–39
10. Soga, M., Kato, T., Ohta, M., Ninomiya, Y.: Pedestrian detection using stereo vision and tracking. In: Proc. of The 11th World Congress on Intelligent Transport Systems. (2004)

11. Zhao, L., Thorpe, C.E.: Stereo- and neural network-based pedestrian detection. IEEE Trans. on ITS **01** (2000) 148–154
12. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision **63** (2005) 153–161
13. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. on SMC **34** (2004) 334–352
14. Bertozzi, M., Broggi, A., Fascioli, A., Tibaldi, A., Chapuis, R., Chausse, F.: Pedestrian localization and tracking system with kalman filtering. In: Proc. of IEEE Intelligent Vehicles Symposium. (2004) 584–589
15. Isard, M., Blake, A.: Condensation—conditional density propagation for visual tracking. International Journal of Computer Vision **29(1)** (1998) 5–28
16. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3D human figures using 2D image motion. In: Proc. of ECCV. (2000) 1–18
17. Sigal, L., Bhatia, S., Roth, S., Black, M.J., Isard, M.: Tracking loose-limbed people. In: Proc. of CVPR. Volume 1. (2004) 421–428
18. Balan, A.O., Sigal, L., Black, M.J.: A quantitative evaluation of video-based 3D person tracking. In: Proc. of IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. (2005) 349–356
19. Kitagawa, G.: Monte-carlo filter and smoother for nongaussian nonlinear state space models. Journal of Computational and Graphical Statistics **5(1)** (1996) 1–25
20. Ashida, J., Miyamoto, R., Tsutsui, H., Onoye, T., Nakamura, Y.: Probabilistic pedestrian tracking based on a skeleton model. In: Proc. of ICIP. (2006) to appear.
21. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1) (1997) 119–139
22. Miyamoto, R., Ashida, J., Tsutsui, H., Nakamura, Y.: Skeleton based stochastic pedestrian tracking for surveillance. In: Proc. of The 10th WMSCI. Volume V. (2006) 206–211
23. Matsumoto, M., Nishimura, T.: Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Trans. on Modeling and Computer Simulation **8** (1998) 3–30

Unsupervised Texture Segmentation Based on Watershed and a Novel Artificial Immune Antibody Competitive Network

Wenlong Huang and Licheng Jiao

Institute of Intelligent Information Processing, Xidian University, 710071Xi'an, China
yykjhuang@sohu.com

Abstract. This paper presents a novel texture segmentation scheme based on two techniques: watershed and a novel structural adaptation artificial immune antibody competitive network (SAIANet). The proposed scheme first partitions image into a set of regions by watershed algorithm and then clusters the watershed regions by SAIANet, where the gray level co-occurrence matrix and the wavelet frame texture features are extracted from each watershed region as the antigens of SAIANet. A new immune antibody neighborhood and an adaptive learning coefficient are presented, and inspired by the long-term memory in cerebral cortices, a long-term memory coefficient is introduced into the network. The minimal spanning tree in graph theory is used to automatically cluster antibody obtained in the output space without a predefined number of clustering. Finally, the presented SAIANet is devoted to performing a fully unsupervised texture segmentation with a superior performance, which makes full use of the watershed segmentation results.

Keywords: Texture segmentation, watershed, structural adaptation, artificial immune network, minimal spanning tree.

1 Introduction

Both neural networks and immunity-based systems are biologically inspired techniques that have the capability of identifying different patterns. They use learning, memory, and associative retrieval to solve recognition and classification tasks. In the recent years, there has been growing interest in using intelligent approaches such as neural network, evolutionary method, and their combined technologies[1]. This paper aims at the combination of the artificial immune system(AIS)[2] and self-organizing feature map neural network(SOFM)[3], to design a novel network model, named structural adaptation artificial immune antibody competitive network(SAIANet), which can use the characteristic knowledge for clustering problem, consequently perform texture segmentation. In SAIANet, a new immune antibody neighborhood and an adaptive learning coefficient are presented, and inspired by the long-term memory in cerebral cortices[4], a long-term memory coefficient is introduced into the network. The model can adaptively map input data into the antibody output space, which has a better adaptive net structure.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 495 – 503, 2006.

As the total number of pixels in the original feature image is usually huge, which can not be used as antigens (train pattern) in SAIANet, a two-stage image segmentation algorithm is proposed. We first segment the original image by the watershed segmentation algorithm. Then, a set of features, including gray co-occurrence matrix and wavelet frame texture features, are extracted from each watershed region, which is regarded as the antigens and inputted into SAIANet. Finally, we utilize the minimal spanning tree in graph theory to automatically cluster antibody obtained in SAIANet.

2 Watershed Segmentation Algorithm

In this work, we use the well-known watershed segmentation algorithm[5][6] to partition an image into nonoverlapping regions. Watershed segmentation is an efficient, automatic, and unsupervised segmentation method. Pixels in a watershed region are homogeneous in the feature space. We then introduce the basic concept of watershed segmentation as follows.

In a natural image, ideal step edges do not often exist since every edge is blurred to some contents. A blurred edge can be modeled by a ramp. For a ramp edge, a usual gradient operator will generate a slope of the edge. Thus, the ramp edge cannot be separated from noise if the slope of the edge is small. Wang proposed a multi-scale gradient operator to solve the above problem [6]:

$$MG(f) = \frac{1}{n}\sum_{i=1}^{n}\left[\left((f \oplus B_i) - (f \Theta B_i)\Theta B_{i-1}\right)\right] \tag{1}$$

where $\oplus$ and Θ denote dilation and erosion, respectively, and B_i is called structural element of size $(2i-1)\times(2i-1)$, and f is the original image.

In the watershed segmentation algorithm, two parameters, *r* and *h*, need to be assigned. Parameter *r* is the size of the structural element of the dilation operators. By using the dilation with the structural element, local minima which size is less than *r* pixels will be eliminated. Besides, parameter *h* is the height of elevation used for removing the local minima with low contract. These two parameters can be used to control the coarseness of the segmentation results. As *r* and *h* increase, the number of regions generated decreases.

If the watershed regions are too large, one big region may contain more than one focused subject in the image, texture feature in the region may not be homogeneous. While if the watershed regions are too small, the computational complexity will increase. In our design, we assign two parameters: *r*=4 and *h*=4. Using this setting, the number of watershed regions is about 1400 for each image.

3 Structural Adaptation Artificial Immune Antibody Competitive Network

In SOFM, a Hebbian learning rule gives an explicit mechanism to adjust the connection strengths among a fixed set of formal neurons. This self-regulated ongoing change in synaptic connectivity endows the SOFM neural network with a dynamic and metadynamic characteristic that can be explored in the realm of immune networks. Inspired by ideas from SOFM and immunology, a novel immune network, i.e. SAIANet, is proposed.

The SAIANet algorithm is summarized in the pseudocode presented below.

```
1. Initialize randomly antibodies in the network and define the parameters: $\eta(1)$, $\beta$, $\gamma$ and $\sigma$. The number of initial antibodies could be set between 0.005*N to 0.01*N, where N is the number of input data points.
2. While not reached the convergence criterion do:
  2.1. For each input pattern do:
    2.1.1. Present all the antigens to the network;
    2.1.2. Calculate the Euclidean distance between the antigens and the antibodies in the network;
    2.1.3. Calculate the neighborhood of each antibody and find the winner antibody;
    2.1.4. Update the weights of the antibodies using the intra-points of their neighborhoods;
  2.2. If (Iteration > $\gamma$) then $\eta(k) = \eta(k)\exp(-\sigma(k-\alpha))$, where k is the current Iteration;
  2.3. If (iteration is multiple of $\beta$) then clone the winner if necessary;
  2.4. If the concentration level of a given antibody is smaller a given threshold that can be set to 0.01*N, then it is pruned from the network..
3. Use the MST criterion proposed to automatically segment the antibodies at the output of the network.
```

3.1 Antibody Neighborhood

An antigen is recognized involves finding the most similar antibody to the given antigen, which is expressed through the equation(2), $Ag_j^{Ab_i}$ shows that the jth antigen is recognized by the ith antibody. This antibody is said to have the highest affinity with the antigen. This is aimed at calculating the antibody neighborhood, i.e. the concentration level of each antibody, what corresponds to the number of antigens recognized by each antibody. The neighborhood of the ith antibody $Ab_{i-}Ag_{Nc}$ is expressed through the equation(3).

$$Ag_j^{Ab_i} = \left\{ Ab_i \middle| \max_i \left(1 / \left\| Ag_j - Ab_i \right\| \right), Ab_i \in Ab_{set}(k) \right\} \tag{2}$$

$$Ab_{i-}Ag_{Nc} = \left\{ Ag_j^{Ab_q} \middle| q = i, Ag_j \in Ag_{set} \right\} \tag{3}$$

where $Ab_{set}(k)$ is the current antibody set, Ag_{set} is the antigen set.

3.2 Antibody Competitiveness Rule

During the competitive phase, the realization of antibody competitiveness is to choose the most stimulated antibody (the winner antibody Ab_w). It is based on the concentration of antigens recognized by an antibody, i.e. the antibody neighborhood, which is expressed by the equation (4).

$$Ab_w = \left\{Ab_i \middle| \max\left(Ab_i^{Concentration}\right),\ Ab_i \in Ab_{set}(k)\right\} \tag{4}$$

$$Ab_i^{Concentration} = sumnum(Ab_{i\_}Ag_{Nc}) \tag{5}$$

where $Ab_i^{Concentration}$ is the concentration of the ith antibody, $sumnum(\cdot)$ is a count operator.

3.3 Antibody Clone Operator

In SAIANet, network growing is inspired by the clonal selection principle, where the most stimulated cell is selected for cloning. The choice of Ab_w is based on the affinity to the antigen, determined during the competitive phase. The antibody with the highest antigen concentration will generate a single offspring (clone) and the two antibodies might turn into memory antibodies after their maturation phase. Whether the winner antibody is cloned is also decided by an affinity threshold ε. Provided that the network growing process is executed every β iterations, the clone process is described in Equation (6).

if $round(k,\beta) = 0$, and $Af_{Agl}^{Abw} > \varepsilon$, then $Ab_{set}(k) = Ab_{set}(k) + Ab_w$,

else $Ab_{set}(k) = Ab_{set}(k)$ (6)

where $Af_{Agl}^{Abw} = 1/\|Ab_w - Ag_l\|$, Ag_l is the antigen with the lowest affinity to Ab_w.

3.4 Antibody Death Operator

Antibody death operator is to realize the network pruning policy, which is defined as follows: if a cell p has its concentration level less than a presented value, for example one, longer than a specified length of time, then it can be deleted from the network.

3.5 Antibody Network Learning Rule

The learning rule in SAIANet is similar to the procedure used in SOFM neural networks. Equation (7) shows the weight updating rule used. Thus, antibodies are constantly being moved in the direction of the recognized antigens, and antibody is only adjusted with those antigens in its neighborhood.

$$Ab_i(k+1) = Ab_i(k) + \eta(k)(Ab_{i\_}Ag_{Nc}(p) - Ab_i(k)) \tag{7}$$

where $\eta(k)$ is the learning coefficient, $Ab_{i\_}Ag_{Nc}(p)$ is the pth antigen that is recognized by the ith antibody.

3.5.1 Learning Coefficient

In the above learning rule, the amount of the change is guided by the learning coefficient $\eta(k)$. The learning coefficient $\eta(k)$ is set a large enough value in the beginning of learning, then after α iterations, it is exponentially decreased by a factor

σ and is adaptively adjusted with each antibody affinity. So each antibody owns its learning coefficient which is different from the others. This process is described in equation (8) and (9).

$$\eta(k) = \eta(k)\exp(-\sigma(k-\alpha)) \tag{8}$$

$$\eta^{Ab_i}_{Ab_i\_Ag_{Nc}(p)} = \eta(k)(Af^{Ab_i}_{Ab_i\_Ag_{Nc}(p)} - Af^{Ab_i}_{\max}) \Big/ (Af^{Ab_i}_{\min} - Af^{Ab_i}_{\max}) \tag{9}$$

where $Af^{Ab_i}_{Ab_i\_Ag_{Nc}(p)} = 1/\|Ab_i\_Ag_{Nc}(p) - Ab_i\|$, $Af^{Ab_i}_{\max} = \max\left\{Af^{Ab_i}_{Ag(p)}, Ag(p) \in Ab_i\_Ag_{Nc}\right\}$, $Af^{Ab_i}_{\min} = \min\left\{Af^{Ab_i}_{Ag(p)}, Ag(p) \in Ab_i\_Ag_{Nc}\right\}$

3.5.2 Long-Term Memory Coefficient

In this paper, based on the biological aspects[4], we develop a novel long-term memory model. The formation strategy for long-term memory is that if an input pattern is something new then the pattern can be stored as long-term memory. With this strategy, we design a long-term memory $y_i(k)$ by utilizing the affinity of antibodies, which is expressed through the equation (10).

$$y_i(k) = \exp\left(-\frac{\|Ab_i(k) - Ab_d(k)\|^2}{2\delta_e(e_m)^2}\right) \tag{10}$$

where $\|Ab_i(k) - Ab_d(k)\|^2$ is the distance between the ith antibody Ab_i and the antibody Ab_d, which is the nearest antibody to Ab_i in the current antibody set. If $\|Ab_i(k) - Ab_d(k)\|^2$ is relatively large, then the antibody Ab_i may be something new and will have a chance to recognize more antigens, so it has a tendency to be stored as long-term memory, otherwise the input pattern will disappear or be replaced by the other antibodies rapidly.

With the adjusted learning coefficient $\eta(k)$ and long-term memory coefficient $y_i(k)$, we have the new complete antibody learning rule given as

$$Ab_i(k+1) = Ab_i(k) + \eta^{Ab_i}_{Ab_i\_Ag_{Nc}(p)} y_i(k)(Ab_i\_Ag_{Nc}(p) - Ab_i(k)) \tag{11}$$

3.6 Convergence Criterion

The convergence criterion is used to check the stability of the network topology. It is assumed that the network topology has reached stability if during the last $5*\beta$ iterations there was no variation in the number of antibodies.

3.7 Defining the Number of Clusters

We propose the use of a minimal spanning tree (MST) [10], which is a tool from graph theory for clustering. The MST defines a neighborhood relationship among antibodies and determines the number of clusters found by the learning algorithm. An inconsistent edge may be determined as follows: if the length of an edge is greater than the average plus two standard deviations, then this edge is considered inconsistent [1].

4 Experimental Results

We have demonstrated the procedure of our texture segmentation scheme above. It mainly consists of three parts: watershed segmentation, feature extraction and pixels clustering. In this paper, we used two kinds of the texture features, including the gray level co-occurrence matrix(GLCM)[7] and the wavelet frame texture features[8][9]. The feature vector for GLCM had a set of 12 features, while the wavelet frames based technique had 10 features. The antigens, i.e. input data, are the mean of the feature vectors of the pixels in each watershed region. In this section, we make some experiments to assess the performance of the proposed texture segmentation scheme. The performance of the proposed method is compared with that of the FCM algorithm[11]. Three natural and one SAR texture segmentation examples are presented in the following. All the natural textured image is from the Brodatz album[12]. The parameters used to run SAIANet in all experiments to be reported here were: $\beta = 2$, $\alpha = 70$, $\eta(1) = 0.3$, $\sigma = 0.02$, $\varepsilon = 0.03$, unless otherwise specified.

The first example is shown in Fig.1**.** The original textured image shown in Fig.1(a) which is a composite of two textures. The watershed segmentation is illustrated in fig.1(b), which is regarded as antigens in our SAIANet algorithm. The segmentation results of FCM and SAIANet are illustrated in Fig.1(c) and Fig.1(d), respectively. We observed that the two regions are well segmented by these two algorithms. However, with the FCM algorithm, the segmentation result still has some noise, while the result used by SAIANet algorithm is less speckled and smoother. The segmentation accuracy of SAIANet algorithm is 98.92%, and that of FCM is 97.54%.

The next two examples are two and three textures in the original images, which are shown in Fig.2 and Fig.3 , respectively. The FCM segmentation results are displayed in (b), and the watershed segmentation and the SAIANet segmentation results are displayed in (c) and (d) , respectively. It is obviously that SAIANet segmentation result is better than FCM, either in the boundary localization or the noise smoothness.

The last exmple is a SAR image which is a more challenging case. The image is composed of three parts: water, grass and building. The processing of the SAR texture image is generally more involved, mainly because the textures appearing in the SAR image are usually nonstationary. Fig.4(a) to (d) show the segmentation results by the watershed, FCM and SAIANet. It is clearly seen that our segmentation result is much closer to the ground truth. The result of SAIANet is more homogeneous and smoother than that of the FCM algorithm, moreover, there are many error segmentations between water and building in the latter, which again indicates our method is effective and robust to noise.

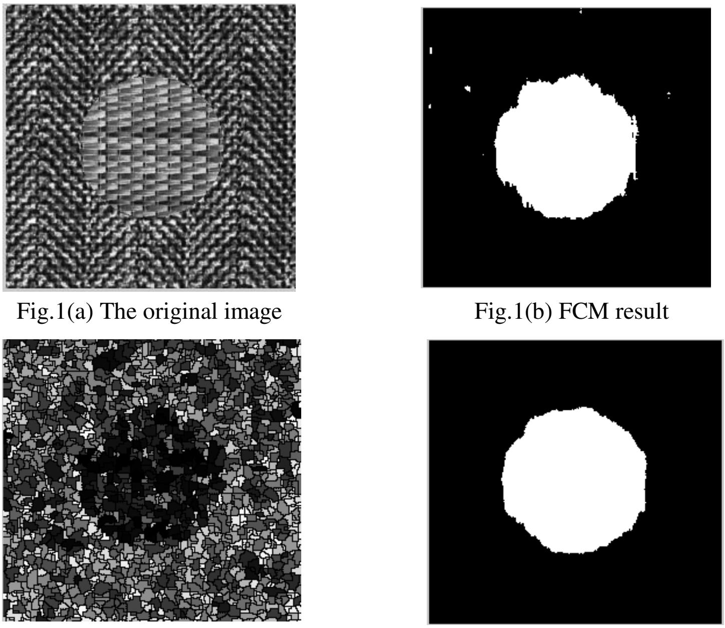


Fig.1(a) The original image

Fig.1(b) FCM result

Fig.1(c) watershed segmentation

Fig.1(d) SAIANet result

Fig. 1. Segmentation results

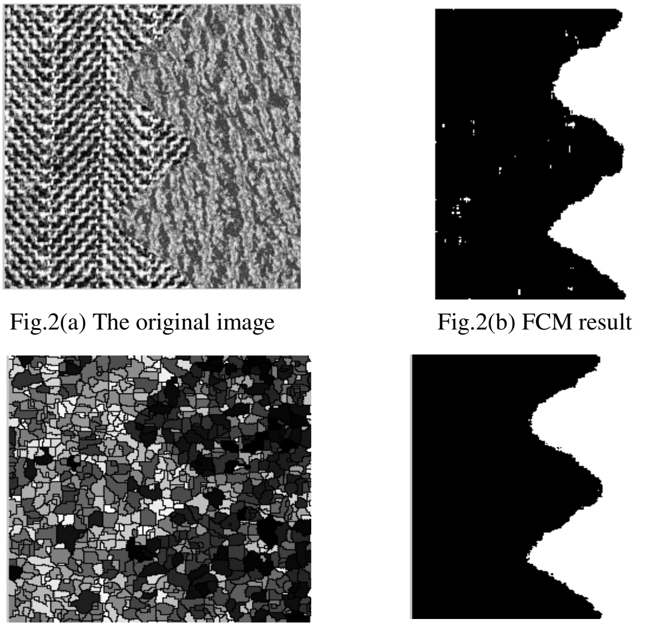


Fig.2(a) The original image

Fig.2(b) FCM result

Fig.2(c) watershed segmentation

Fig.2(d) SAIANet result

Fig. 2. Segmentation results

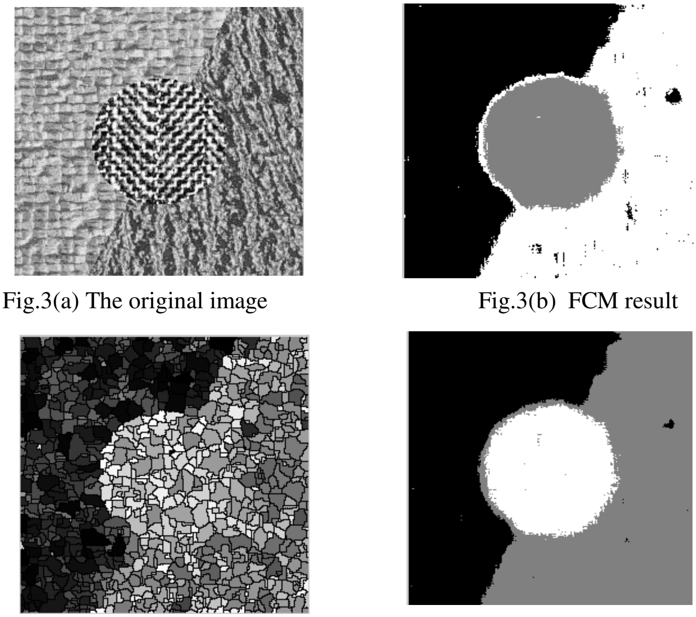


Fig.3(a) The original image

Fig.3(b) FCM result

Fig.3(c) watershed segmentation

Fig.3(d) SAIANet result

Fig. 3. Segmentation results

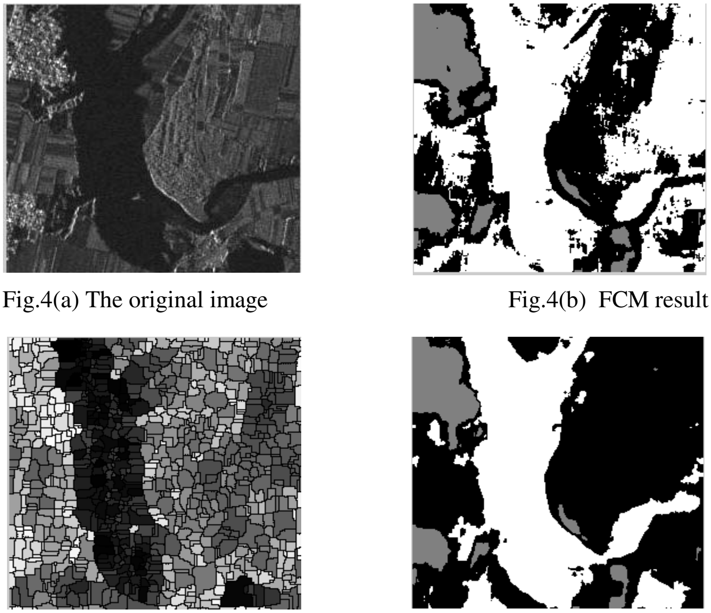


Fig.4(a) The original image

Fig.4(b) FCM result

Fig.4(c) watershed segmentation

Fig.4(d) SAIANet result

Fig. 4. Segmentation results

5 Conclusion

This paper has presented a new fully unsupervised segmentation algorithm based on watershed and an artificial immune antibody competitive network. The immune antibody network is derived from vertebral animal immune system, SOFM neural network and the long-term memory in cerebral cortices, which has a better adaptive net structure. With the antigens get from each watershed region, the immune network can automatically cluster antibody obtained in the output space without a predefined number of clustering. The experimental results show our texture segmentation system has an excellent segmentation performance, and should be useful in many practical applications, on account of its high segmentation accuracy, its flexibility and its robust property.

Acknowledgements. This work is supported by grants from the Key Program of National Natural Science Foundation of P.R.China (No. 60133010 and 60372045) and the National Grand Fundamental Research 973 Program of China under Grant No.2001CB309403.

References

1. Helder, K., L. N., de Castro, Fernando, J., Von Zuben, RABNET: A Real-Valued Antibody Network for Data Clustering[C], GECCO'05, Washington, DC, USA. June 25-29, 2005, 371-372.
2. Hunt, J. E. and Cooke, D. E.. Learning Using an Artificial Immune System[J], Journal of Network and Computer Applications, 1996(19),189 – 212.
3. Kohonen, T., The self-organizing map[J], Neurocomput., vol. 21, 1998,1–6.
4. Lanprecht, R. and LeDoux, J., Structura1 plasticity and memory[J], Nature Rev. Neuroscience, vol. 5, 2004,45-54.
5. Vincent, L., Soille, P., Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. IEEE Transactions on PAMI. Vol. 13(6) , Jun.1991, 583-598.
6. Wang, D., A Multiscale Gradient Algorithm for Image Segmentation Using Watersheds. Pattern Recognition. 30(12) ,1997, 2043-2052.
7. Gotlieb, C. C. and Kreyszig, H. E., Texture descriptors based on cooccurrence matrices[J], *Comput. Vision, Graph. Image Processing,* vol. 51, 1990, 70-86.
8. Unser, M., 1995. Texture classification and segmentation using wavelet frames. IEEE Trans. Image Process. Volume 4, Issue 11, Nov. 1995,1549-1560.
9. Laine, A., Fan, J., Frame representation for texture segmentation. IEEE Trans. Image Process. Volume 5, Issue 5, May, 1996,771-780.
10. Leclerc B, Minimum spanning trees for tree metrics: abridgements and adjustments[J], Journal of Classification, 1995(12), 207-241.
11. Bezdek, J. C., Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum Press, 1981.
12. Brodatz P. Textures: A Photographic Album for Artists and Designers. New York, Dover Publication, 1966.

A Novel Driving Pattern Recognition and Status Monitoring System

Jiann-Der Lee, Jiann-Der Li, Li-Chang Liu, and Chi-Ming Chen

Department of Electrical Engineering,
Chang-Gung University, Tao-Yuan, Taiwan
jdlee@mail.cgu.edu.tw, m9321035@stmail.cgu.edu.tw,
Lichang_Liu@alumni.pitt.edu, yumpy.chen@msa.hinet.net

Abstract. This paper describes a novel driving pattern recognition and status monitoring system based on the orientation information. Two fixed cameras are used to capture the driver's image and the front-road image. The driver's sight line and the driving lane path are found from these 2 captured images and are mapped into a global coordinate. Two correlation coefficients among the driver's sight line, the driving lane path and the car heading direction are calculated in the global coordinate to monitor the driving status such as a safe driving status, a risky driving status and a dangerous driving status. The correlation coefficients between the lane path and car heading direction in a fixed period are analyzed and recognized as one of 4 driving patterns by HMM. Four driving patterns including the driving in a straight lane, the driving in a curve lane, the driving of changing lanes, and the driving of making a turn are able to be recognized so far.

Keywords: Driving event, HMM, Ada-boosted, driving safety monitoring, intelligent transport, and shape-context.

1 Introduction

Car accident deaths in Taiwan increase in the last years and become one of top 10 accident death factors, so intelligent transportation systems have became an important research aspect. Many intelligent transportation and pre-crash systems [1][2][3][4][5] have been proposed to avoid a potential car accident and to increase driving safety. Mitrovic used hidden Markov models (HMMs) to develop a driving event recognition system [6]. Numeric data were acquired from the longitudinal and lateral beam equipments and a GPS receiver to measure the average speed and acceleration information which were converted into symbols to train HMM. His system is able to recognize hundreds of driving events such as "Stop", "U Turn", "Left Turn", "Left Turn on RA", etc. To implement an easier driving event recognition system, we focus on using image information captured from webcams or video cameras because an image-based system is cheaper and easier to be setup. In this paper, an intelligent

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 504 – 512, 2006.

transport system which is able to recognize the current driving pattern and to monitor the driving status is introduced. Two fixed cameras were used in the proposed system to capture the driver's image and the front-road image, and the orientation information including the car heading direction, the driver's sight line and the lane path from the 2 captured images were mapped into a global coordinate to calculate 2 correlation coefficients. Soft-shape-context concept [7] was used in the correlation coefficient calculation to indicate the orientation similarity relations which also provided the driver's steering degrees. The lane correlation coefficient was between the lane path and car heading direction, and the driver correlation coefficient to indicate the driver's cognition was between the lane path and the driver's sight line. The sequences of tens of the lane correlation coefficients were used for HMM training and identification of 4 driving patterns such as driving in a straight lane and making a turn. The lane and driver correlation coefficients were used for the driving status monitoring which was recognized as one of 3 stages: safe, risky and dangerous stages. The proposed system is able to cooperate with other pre-crashed systems based on the distance information to neighbor cars, too.

2 The Details of the Proposed System

The proposed system contains 3 major parts: the orientation information calculation part, the driving pattern recognition part, and the driving status monitoring part. Two fixed cameras were used in the proposed system to capture 2 images: the driver's image and the front-road image. In the orientation information extraction/calculation part, the image processing technologies were applied on 2 captured images to extract the driver's sight line, the lane path and the car heading direction. These 3 orientations were mapped into a global coordinate and calculated for 2 correlation coefficients: the lane correlation coefficient and the driver correlation coefficient, by using Soft-shape-context concept. In the driving pattern recognition part, the sequences of the lane correlation coefficients of different driving events were used for HMM training which was used to identify the current driving event. In the driving status monitoring part, the lane and driver correlation coefficients were used to monitor the driver' steering degrees and were classified as 3 steering stages: safe, risky, and dangerous stages according the coefficient values. The driver's sight line is used in the driver correlation coefficient calculation because the driver's sight line information shows the driver's conscious steering to the cases of changing lanes and making a turn. The system flowchart is shown in Fig. 1.

2.1 The Extraction of Three Orientations and the Calculation of Two Correlation Coefficients

Two fixed cameras were placed according planned positions in a global coordinate to capture 2 images: the driver' image and the front-road image. Therefore, the distance

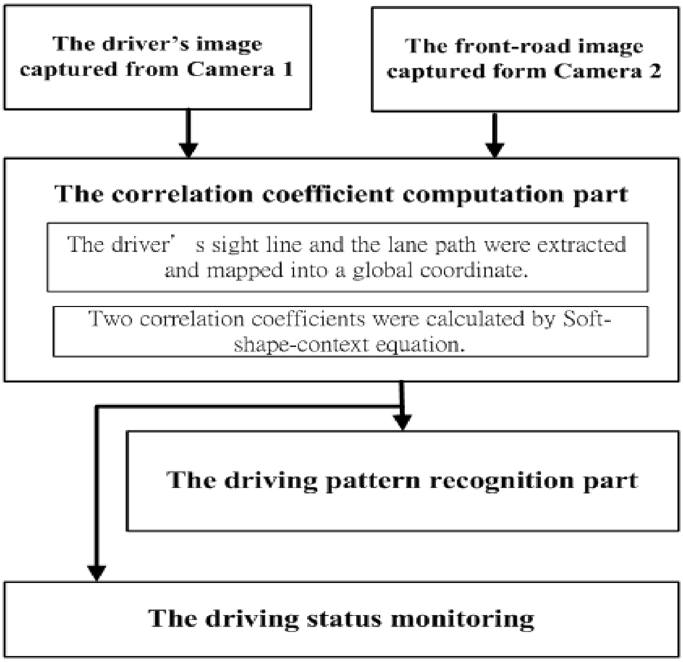

The driver's image captured from Camera 1
The front-road image captured form Camera 2
The correlation coefficient computation part
The driver' s sight line and the lane path were extracted and mapped into a global coordinate.
Two correlation coefficients were calculated by Soft-shape-context equation.
The driving pattern recognition part
The driving status monitoring

Fig. 1. The system flowchart

and line information in these 2 images were able to be mutually fused in the global coordinate domain. The block diagram of the correlation coefficient computation part is shown in Fig. 2.

Ada-boosted algorithm [8][9] was applied twice on the driver's image to extract the driver's face first and to extract the lip position second from the previous extracted face region. From the upper neighbor region of the lips, the driver's philtrum was detected by the binary projection information. The ratio R_{LIP} of two lengths L_3 and L_4 between the fringes of the driver' lips and philtrum as shown in Fig. 3 was calculated by (1) to indicate the driver's sight line. The threshold values of R_{LIP} were 0.7, 0.5, 0.3, -0.7, -0.5, -0.3 to the cases of 12°, 24°, 36°, -12°, -24°, and -36° respectively.

$$R_{LIP} = sign(L_3 - L_4)\frac{\min(L_4, L_3)}{\max(L_4, L_3)} \tag{1}$$

The camera denoted as Camera 2 in Fig. 2 to capture the front-road image was located close to the central rear-view mirror, i.e. the global coordinate was based on Camera 2 and the central vertical line of the global coordinate was the car heading direction. The binary detected results of the red, yellow and white color-segmented points and edge-detection points were mapped into the global coordinate by using the inverse perspective mapping [10] and were added with the previous-frame mapped points. A morphology operator, a thinning operator and Hough transform were

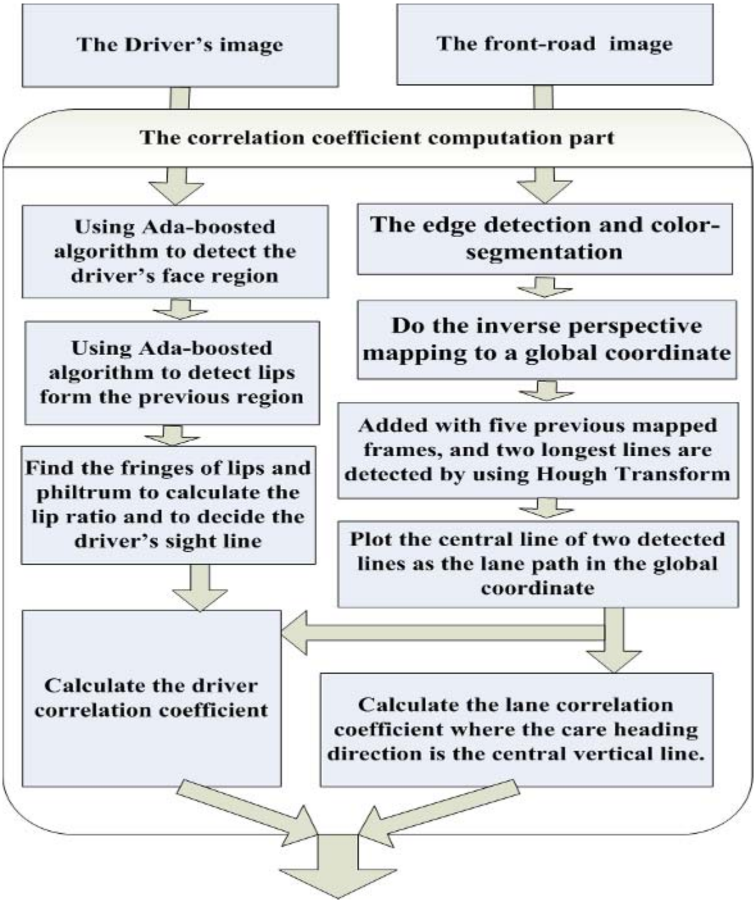

The Driver's image
The front-road image
The correlation coefficient computation part
Using Ada-boosted algorithm to detect the driver's face region
Using Ada-boosted algorithm to detect lips form the previous region
Find the fringes of lips and philtrum to calculate the lip ratio and to decide the driver's sight line
The edge detection and color-segmentation
Do the inverse perspective mapping to a global coordinate
Added with five previous mapped frames, and two longest lines are detected by using Hough Transform
Plot the central line of two detected lines as the lane path in the global coordinate
Calculate the driver correlation coefficient
Calculate the lane correlation coefficient where the care heading direction is the central vertical line.

Fig. 2. The block diagram of the correlation coefficient calculation part

orderly applied on the added results to detect two longest lines close to the central vertical line as the possible traffic lane lines. Therefore, the lane path was the central line of detected longest lines.

After that the driver's sight line, the lane path were found in the global coordinate and the car heading direction was defined as the central line, Soft-shape-context concept [7] was used to calculate the lane correlation coefficient and the driver correlation coefficient. The shape context [11] is a similarity measurement to measure two shapes in two images by accumulation of counting differences in each bin. Edge-detection points of one shape are accumulated by their locations in bins to generate a bin histogram denoted as $h_i(k)$ where k is the k^{th} bin to the i^{th} image. The correlation coefficient between the i^{th} image and the j^{th} image is defined in (2).

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k)-h_j(k)]^2}{h_i(k)+h_j(k)} \tag{2}$$

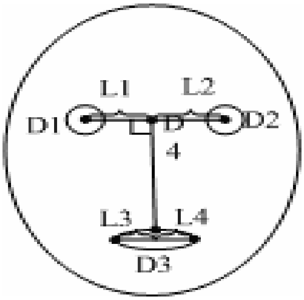
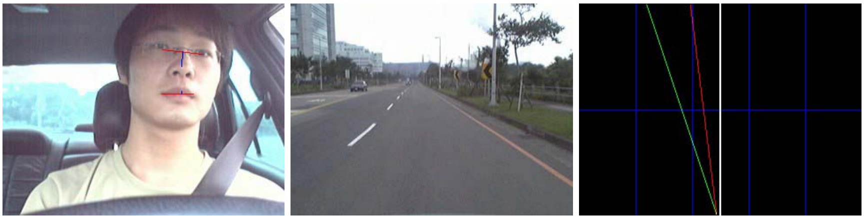


Fig. 3. The face model

Smaller values of correlation coefficients indicate that two shapes are more similar. The soft-shape context method [7] utilizes a low-pass filter on smoothing the bin histogram to avoid the sensitiveness around the bin boundaries. Therefore, the lane correlation coefficient between the lane path and the car heading direction and the driver correlation coefficient between the driver's sight direction and the lane path in a global bin map were calculated by using the soft-shape context concept.

The lane correlation coefficient C_{Lane} indicates the driving consistency and the driver correlation coefficient C_{Driver} indicates the driver's cognitive degrees. In Fig. 4, an example was shown. Ten equal rectangular bins were separated by blue lines, the detected lane path was in the red color, the driver's sight line was in the green color, and the car heading direction was the central vertical line in the white color. C_{Lane} is 5.54 and C_{Driver} is 236.43 in Fig. 4 because the driver looked at its left side to increase C_{Driver} values.



(a) (b) (c)

Fig. 4. An example of the detected lane path mapped into a 10-rectangular-bin map in a global coordinate. (a) the captured driver's image, (b) the captured front-road image, and (c) the detected lane path in red color and driver's sight line in green color in the bin map.

2.2 The Driving Pattern Recognition Part

Lane correlation coefficients were used in the driving pattern recognition part. A sequence contained tens of lane correlation coefficients, and sequences of a same driving event were trained for HMM. Four driving patterns/events: driving inside a straight lane, driving in a curve lane, making a turn and changing lanes were trained so far. The distribution of the lane correlation coefficients was used as the feature to distinguish the driving events. The lane correlation coefficients of the driving inside a straight lane were small and as almost a straight line in the distribution map. The lane coefficients of the driving in a curve lane were high and formed a straight line. The lane correlation coefficients of the case for changing lanes changes from small values at the beginning, then into high values, and finally to small values again. Therefore, from the distributions of the lane correlation coefficients, the driving events were distinguished.

2.3 The Driving Status Monitoring Part

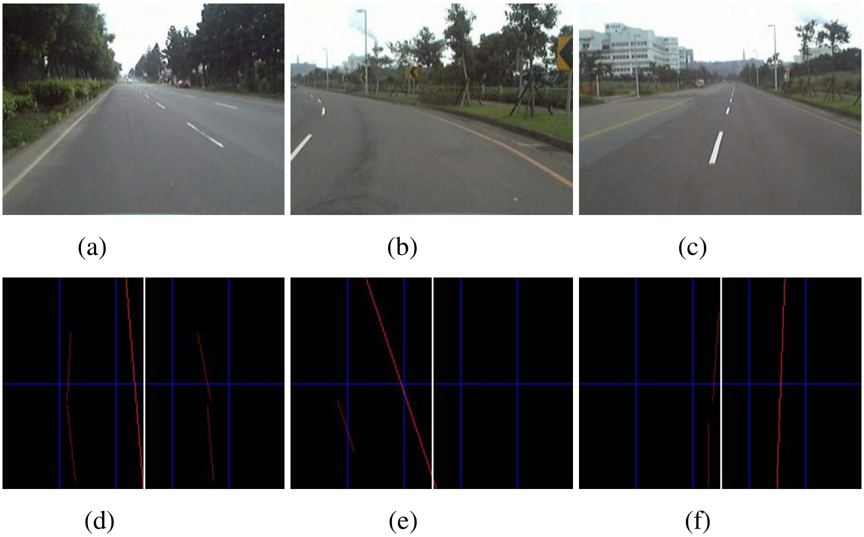
The driver and lane correlation coefficients were used to monitor the driving status which was judged as one of the safe, risky and dangerous stages in this part. The driver correlation coefficient was used to verify the lane correlation coefficient, i.e. a high value of the lane correlation coefficient doesn't always indicate a dangerous situation since this driving event may be still under the driver's control. For example, the driver correlation coefficient should reduce the danger degrees in the case when car was shifted to the left lane and the driver looked at the left side to show that this behavior was under the driver's steering. A danger coefficient defined in (3) utilized the lane correlation coefficient and the driver correlation coefficient to monitor the current driving status. Smaller values of C_{Danger} indicate a safer driving status.

$$C_{Danger} = C_{Lane} \cdot C_{Driver} \tag{3}$$

3 Experimental Results

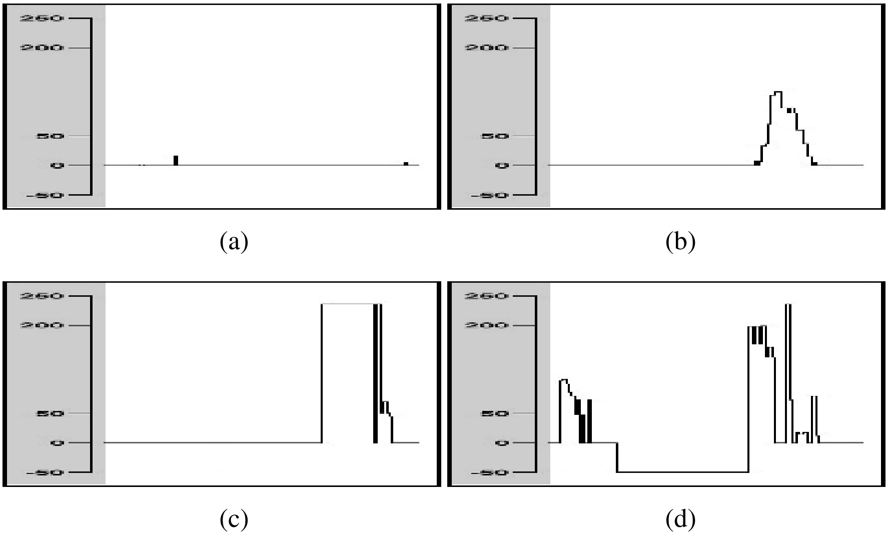
Four sequences including driving in a straight lane, driving in a curve lane, making a right turn, making a left turn and changing lanes were tested but only some were shown in this section because of the page limitation. Three original front-road images including 3 driving situations of driving in a straight lane, driving in a curve lane, and changing lanes were shown in Fig. 5(a) to 5(c) respectively, and their lane-path bin maps in the global coordinate were shown in Fig. 5(d) to 5(f) respectively. The car heading direction was the white line and the lane path was the red line in a 10-bin map as in Fig. 5(d) to 5(f). The C_{Lane} coefficients were 0, 86.6, and 236.3 in Fig. 5(d) to 5(f) respectively.

The typical C_{Lane} distributions of 4 driving patterns were shown in Fig. 6(a) to 6(d). All C_{Lane} coefficients of driving in a straight lane were below 50 as in Fig. 6(a), and all C_{Lane} coefficients of driving in a curve lane were between 60 and 150 for a period as in Fig. 6(b). The C_{Lane} coefficients of changing lanes were small at the beginning , increased fast during the shifting, and decreased to small values when drove in another lane as shown in Fig. 6(c). The C_{Lane} coefficients of making a right or left turn in Fig. 6(d) were assigned into negative values when the car was in the intersection and the lane path was not able to be detected. HMM was trained by using distribution information to recognize four driving patterns.


(a) (b) (c)

(d) (e) (f)

Fig. 5. Three examples of lane correlation coefficients. (a) the front-road image of driving in a straight lane, (b) the front-road image of driving in a curve lane, (c) the front-road image of changing lanes, (d) the lane path of (a) in the bin map in the global coordinate, (e) the lane path of (b) in the bin map in the global coordinate, and (f) the lane path of (c) in the bin map in the global coordinate.



(a) (b)

(c) (d)

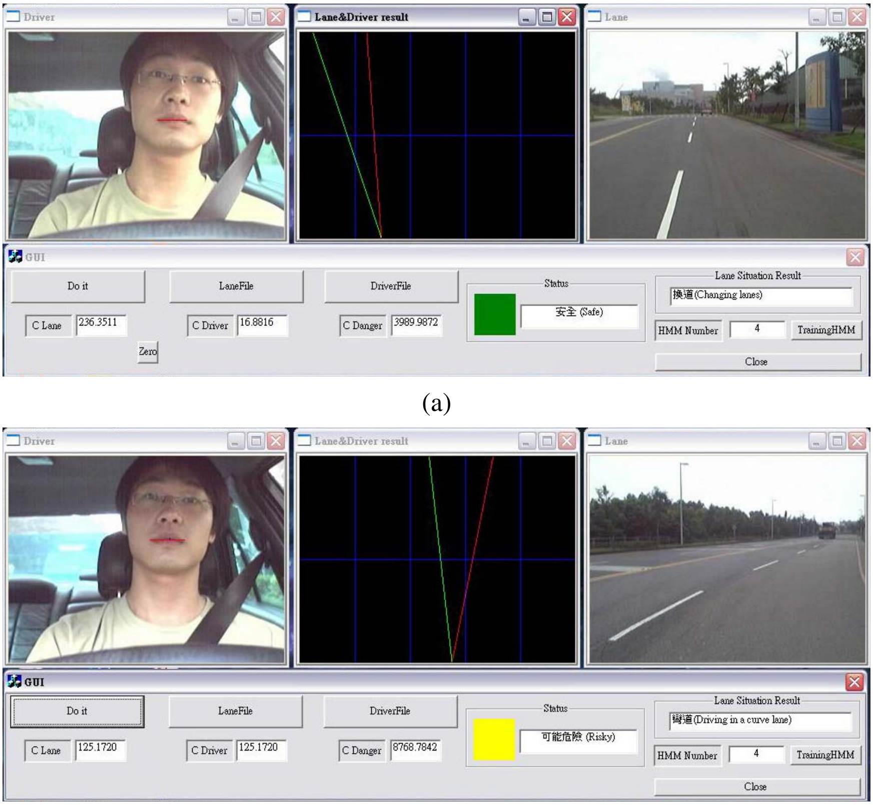
Fig. 6. The typical C_{Lane} distributions of 4 driving patterns. (a) the distribution of driving in a straight lane, (b) the distribution of driving in a curve lane, (c) the distribution of changing lanes, and (d) the distribution of making a right turn.

The possible $C_{Dangery}$ ranges for 3 driving statues were suggested in Table 1.

Table 1. The coefficient ranges of C_{Danger} for three driving statuses

	C_{Dnager}
The safe driving status	3000
The risky driving status	3000 to 24000
The dangerous driving status	Above 24000

Two recognized results were shown in Fig. 7. In Fig. 7(a), the driving status was suggested as the safe status and the recognized driving pattern was the case of changing lanes. In Fig. 7(b), the driving status was suggested as the risky status displayed in the yellow color and the recognized driving pattern was the event of driving in a big curve lane.



C Lane 236.3511

C Driver 16.8816

C Danger 3989.9872

Status: 🟩 安全 (Safe)

Lane Situation Result: 換道(Changing lanes)

HMM Number 4

(a)

C Lane 125.1720

C Driver 125.1720

C Danger 8768.7842

Status: 🟨 可能危險 (Risky)

Lane Situation Result: 彎道(Driving in a curve lane)

HMM Number 4

(b)

Fig. 7. Two recognized results. (a) the driving status was suggested as the safe status and the recognized driving pattern was the case of changing lanes, and (b) the driving status was suggested as the risky status and the recognized driving pattern was the case of driving in a big curve lane.

4 Conclusion and Future Works

A novel intelligent transport system has been presented. In the proposed system, the driver's sight line, the lane path, and the car heading direction were mapped into the global coordinate to calculate the driver and lane correlation coefficients by using the soft-shape-context concept. Sequences of lane correlation coefficients were trained for HMM to recognize four common driving patterns including driving in a straight lane, driving in a curve lane, changing lanes, and making a turn. From combined values of the driver and lane correlation coefficients, one driving status of the safe, risky, and dangerous statuses was suggested. From the experimental results, the driving statuses could be monitored well in the ideal situations. In a practical situation when the car was standing for the green traffic light or a chance to make a left turn, the current system couldn't recognize this driving event well. Some traffic letters and signs painted on the road may affect the lane path detection. In the future, the motion information and the distance information to neighbor cars are considered to be added for the improvement of the monitoring suggestions.

Acknowledgments. This work was supported by National Science Council, R.O.C. with Grant No. NSC 95-3113-p-182-001.

References

1. Chapuis, R., Aufrere, R., Chausse, F.: Accurate Road Following and Reconstruction. IEEE Trans. on Intelligent Transportation System, Vol. 3(4) (2002) 261–270
2. Liu, T., Zheng, N., Cheng, H., Xing, Z.: A Novel Approach of Road Recognition Based on Deformable Template and Genetic Algorithm. Proc. of Intelligent Transportation System Conference, Vol. 2 (2003) 1251–1256
3. Collado, J.M., Hilario, C., Escalera, A. de la, Armingol, J.M.: Detection and Classification of Road Lanes with a Frequency Analysis. IEEE Intelligent Vehicles Symposium, (2005) 78–83
4. Jeong, S.G., Kim, C.S., Lee, D.Y., Ha, S.K. Lee, D.H., Lee, M.H., Hashimoto, H.: Real-time Lane Detection for Autonomous Vehicle. Proc. of IEEE ISIE'01-Industrial Electronics, (2001) 1466–1471
5. Wu, W., Yang, J., Zhang J.:A Multimedia System for Route Sharing and Video-based Navigation. CD-ROM Proc. of IEEE International Conference on Multimedia & Expo (ICME), (2006) 73-76
6. Mitrovic, D.: Reliable Method for Driving Events Recognition. IEEE Trans. on Intelligent Transportation Systems, Vol. 6(2) (2005) 198-205
7. Liu, D., Chen, T.: Soft Shape Context For Iterative Closest Point Registration. IEEE Int. Conf. on ICIP (2004) 1081-1084
8. Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 1 (2001) 511–518
9. Bradski, G., Kaehler, A., Pisarevsky, V.: Learning-based Computer Vision With INTEL's Open Source Computer Vision Library. INTEL Technology Journal, Vol. 9(1) (2005) 119–130
10. Bertozzi, M., Broggi, A., Fascioli, A.: Stereo Inverse Perspective Mapping: Theory and Applications. Image and Vision Computing Journal (1998) 585–590
11. Belongie, S., Malike, J., Puzicha, J.: Shape Matching and Object Recognition Using Shape Contexts. IEEE Trans. Pattern Anal. Mach. Intelligence, Vol. 24(4) (2002) 509–522

Advances on Automated Multiple View Inspection

Domingo Mery and Miguel Carrasco

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile
Av. Vicuña Mackenna 4860(143), Santiago de Chile
dmery@ing.puc.cl

Abstract. Automated visual inspection is defined as a quality control task that determines automatically if a product, or test object, deviates from a given set of specifications using visual data. In the last 25 years, many research directions in this field have been exploited, some very different principles have been adopted and a wide variety of algorithms have been appeared in the literature. However, automated visual inspection systems still suffer from i) detection accuracy, because there is a fundamental trade off between false alarms and miss detections; and ii) strong bottleneck derived from mechanical speed and from high computational cost. For this reasons, automated visual inspection remains an open question. In this sense, *Automated Multiple View Inspection*, a robust method that uses redundant views of the test object to perform the inspection task, is opening up new possibilities in inspection field by taking into account the useful information about the correspondence between the different views. This strategy is very robust because in first step it identifies potential defects in each view and in second step it finds correspondences between potential defects, and only those that are matched in different views are detected as real defects. In this paper, we review the advances done in this field giving an overview of the multiple view methodology and showing experimental results obtained on real data.

Keywords: automated visual inspection, multiple view geometry, industrial applications.

1 Introduction

Visual inspection is defined as a quality control task that determines if a product deviates from a given set of specifications using visual data[1]. Inspection usually involves measurement of specific part features such as assembly integrity, surface finish and geometric dimensions. If the measurement lies within a determined tolerance, the inspection process considers the product as accepted for use. In

[1] For an extended overview of automated visual inspection, the reader is referred to excellent review papers by Malamas et al. [1] and Newman and Jain [2]. The information given in this paragraph was extracted from these papers.

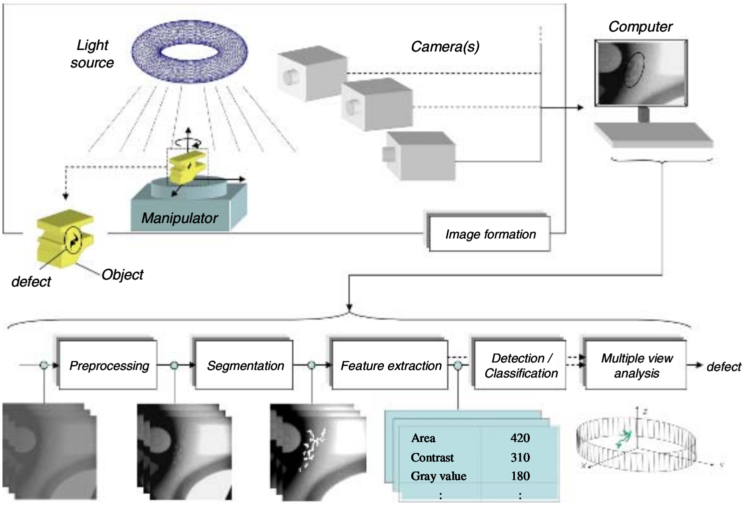
L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 513–522, 2006.

industrial environments, inspection is performed by human inspectors or automated visual inspection systems. Although humans can do the job better than machines in many cases, they are slower than the machines and get tired quickly. Additionally, human inspectors are not always consistent and effective evaluators of products because inspection tasks are monotonous and exhausting, even for the best-trained experts. Typically, there is one rejected in hundreds of accepted products. Moreover, human experts are difficult to find or maintain in an industry, require training and their skills may take time to develop. It has been reported that human visual inspection is at best 80% effective. In addition, achieving human 100%-inspection, where it is necessary to check every product thoroughly to ensure the safety of consumers, typically requires high level of redundancy, thus increasing the cost and time for inspection. For instance, human visual inspection has been estimated to account for 10% or more of the total labor costs for manufactured products. Moreover, in some environments (*e.g.*, underwater inspection, nuclear industry, chemical industry, etc.) human visual inspection may be difficult or dangerous. For these reasons, computer vision has been gradually replacing more and more human inspection.

Comprehensive reviews on automated visual inspection are given in [1,2,3,4]. According to these surveys, approaches developed for automated visual inspection are tailored to the inspection task, *i.e.*, there is no general approach applicable to all cases because the development is an *ad hoc* process. Although there are several approaches that have been developed in the last 25 years, automated visual inspection systems still suffer from i) detection accuracy, because there is a fundamental trade off between false alarms and miss detections; and ii) strong bottleneck derived from mechanical speed and from high computational cost. For this reasons, automated visual inspection remains an open question. In this paper, we present recent advances on *Automated Multiple View Inspection*, a robust method that uses redundant views to perform the inspection task. This novel strategy is opening up new possibilities in inspection field by taking into account the useful information about the correspondence between the different views of the test object. It is very robust because in first step it identifies potential defects in each view and in second step it finds correspondences between potential defects, and only those that are matched in different views are detected as real defects. The paper gives an overview of the multiple view methodology and show experimental results obtained on real data.

2 General Overview of the Multiple View Approach

The principle aspects of an automated multiple view inspection system are shown in Fig. 1. Typically, it comprises the following five steps: i) a manipulation system for handling the test piece (manipulator, robot, etc.), ii) an energy source (light, X-ray, etc.), which irradiates the object under test with, iii) image acquisition system (CCD cameras, image intensifier, etc.) that register digital images of the test piece, and iv) a computer to perform the digital analysis of the images and to classify the test piece accepting or rejecting it.



Area	420
Contrast	310
Gray value	180
:	:

Fig. 1. Computer vision system for automated visual inspection

In the computer-aided inspection, our aim is to identify defects automatically using computer vision techniques. The general automated inspection process, presented in Fig. 1, consists of image formation, preprocessing, segmentation, feature extraction, detection/classification and multiple view analysis [5]. Typically, automated visual inspection using only one view does not follows the last step. The mentioned six steps are explained in further detail:

i) Image formation: Images of the test object are taken and stored in the computer. The human eye is only capable of resolving around 40 grey levels [6], however in automated visual inspection grey level resolution must be a minimum of 2^8 levels. In some applications with X-rays, 2^{16} grey levels are used [7], which allows one to evaluate both very dark and very bright regions in the same image. On the other hand, color image systems are able to capture images in several color spaces with 2^{24} different colors [8,9]. Nowadays, a digital image used in automated visual inspection contains usually more than 2^{20} pixels.
ii) Image preprocessing: The quality of the images is improved using contrast enhancement, noise removal and image restoration techniques. Typically, image enhancement is achieved by histogram manipulation, and noise removal by frame averaging or edge-preserving filtering [6]. Edge-preserving filtering is important for defect detection, because it is desirable to smooth the noise without blurring the edges. Moreover, image restoration involves recovering detail in severely blurred images, which is possible when the causes of the imperfections are known *a-priori* [10,11]. This knowledge may exist as an analytical model, or as *a-priori* information in conjunction with knowledge (or assumptions) of the physical system that provided the imaging process in the first place.

iii) Image segmentation: The digital images are divided into disjoint regions with the purpose of separating the parts of interest from the rest of the scene. Image segmentation plays one of the most important roles in real world computer vision systems. In the last 40 years, this field has experienced significant growth and progress [12][2]. According to [13], monochrome image segmentation techniques are classified into the following categories: histogram thresholding, feature space analysis based methods, edge detection based methods, region based methods, fuzzy logic techniques and neural networks, and pointed out that most of them can be extended to color images by representing color information in appropriate color spaces. However, better performance in segmentation of color images is achieved using vector-value techniques that treat the color information as color vectors in a vector space provided with a vector norm [14]. In image segmentation for detecting defects we aim to separate potential defects from background.

iv) Feature extraction: Since some structural parts of the object could be erroneously segmented as defectively regions, we denoted them as *potential defects*. Subsequently, additional steps are required to eliminate the false alarms of the potential defects. The first of these steps is feature extraction, which is centered principally around the measurement of geometric and chromatic characteristics of regions. Contrast based on crossing line profiles [15] and texture [16] are very helpful to distinct defectively regions from its neighbors. After feature extraction, it is important to know which features provide relevant information about defects. For this reason, a feature selection [17] is performed to find the best subset of the input future set that separates the real defects from the false alarms. Methods based on sequential forward/backward selection achieve effective and fast results but they are suboptimal [18]. On the other hand, a branch and bound method guarantees to find the optimal subset, although the complexity is greater than the mentioned methods, it can be reduced considerably using a fast technique [19].

v) Detection/classification: The extracted (and selected) features of each region are analyzed in order to detect or classify the existing defects. We differentiate between detection and classification of defects. Detection corresponds to a binary classification, because in the detection problem, the classes that exist are only two: 'defects' or 'no-defects', whereas the recognition of the type of defect (*e.g.*, voids, cracks, bubbles, inclusions and slags) is known as classification of defect types [20]. Normally, the 'defect' class constitutes a very small fraction of the total search area. Therefore, the 'defect' class will be either empty or sparsely populated [21]. This implies that there are not sufficient data to train a statistical classifier or statistically evaluate the performance of a detector. In these cases, where the defect probability is very small, minimization of the error probability is not a good criterion of performance, because it can be minimized by classifying every region as 'no-defect' (in a domain where the classes are distributed in a 1:99 ratio (skew = 10^2), the maximum likelihood gives 99%

[2] Only last year, 194 papers with the word 'image' and 'segmentation' in the title field were indexed by the Web of Science of ISI.

accuracy). For this reason, the probability of detection is typically maximized while keeping the probability of false alarms under a certain predefined value (Neyman-Pearson criterion [22]). Other applications with skewed class distribution can be found in fraud detection [23] or target detection in hyperspectral imaging [24]. In order to increase the samples of the defect class, simulation of defects can be used [25]. The classifier is designed using well known pattern recognition techniques, that can be categorize into generative and discriminative approaches [26]. Generative learning focuses on generative description of samples and tend to synthesize configurations from them. Principal component analysis [17], linear discriminant analysis [17] and hidden Markov models [27] are typical generative classifiers that produce a probability density model for pattern recognition. On the other hand, discriminative learning attempts to compute the mapping for classification from input to output directly without modeling the underlying distributions. It normally achieves superior performance than generative approach in many applications. Traditional neural networks [28] and support vector machines [29] are discriminative classifiers that attempt to maximize the classification boundary margin of classes for recognition. Recent research on combining generative and discriminative learning has shown that proper combinations of two models outperforms pure generative or discriminative models [26,30,31].
vi) Multiple view analysis: Multiple view geometry is increasingly being used in machine vision [32]. It describes explicit and implicit models which relates the 3D coordinates of an object to the 2D coordinates of the digital image pixel, the geometric and algebraic constraints between two, three and more images taken at different projections of the object, and the problem of 3D reconstruction from N views. Since in last step certain 'no-defects' could be classified erroneously as 'defects', we use multiple view geometry as a final discrimination step. The key idea is to gain more information about the test object by analyzing multiple views taken at different viewpoints. Thus, the attempt is made to match or track the remaining potential defects along the multiple views. The existing defects can be effectively tracked in the image sequence because they are located in the positions dictated by geometric conditions. In contrast, false alarms can be successfully eliminated in this manner, since they do not appear in the predicted places on the following images and, thus, cannot be tracked. The tracking in the image sequence is performed using algebraic multi-focal constraints: bifocal (epipolar) and trifocal constraints among others [32,33,34]. Multiple view analysis is a useful and powerful alternative for examining complex objects were uncertainty can lead to misinterpretation, because two or more views of the same object taken from different viewpoints can be used to confirm and improve the diagnostic done by analyzing only one image [33,34].

Finally, the performance of an automated visual inspection method is assessed using a validation technique (*e.g.*, cross-validation, bootstrap and jackknife [17,35]). Usually, some of the collected cases are removed before training begins. Then when training is performed, the cases that were initially removed can be used to test the performance of the inspection method on these test data.

Thus, one can evaluates how well the method will inspect the test objects that has not already examined. Confidence intervals, where the true values of the misclassification error is expected to fall, can be obtained from the test sets.

3 Implemented Multiple View Approaches

Automated multiple view inspection was implemented in the quality control of aluminum castings of the automotive industry using X-ray images [5]. However, the methodology can be used in the inspection of other manufactured products. In this section we present two approaches that were implemented using calibrated and uncalibrated image sequences and their results obtained on real data.

i) Calibrated Approach: In [36] we performed the tracking using a calibrated image sequence , *i.e.*, the model 3D → 2D was *a-priori* known because it was obtained in an off-line process called calibration [37]. The calibration of an imaging system is the process of estimating the parameters of the model, which is used to determine the projection of the 3D test object into its 2D digital image. This relationship 3D→2D can be modeled with a transfer function $F: \ R^3 \rightarrow R^2$. Using this model the multi-focal tensors can be calculated in order to evaluate the multi-focal constraints for the correspondences of the potential defects in the image sequence [32]. The calibration was performed using the well-known photogrammetric calibration [38], in which a calibration object whose geometry in 3D space is known with high accuracy. Using this technique a true reconstruction of the 3D space without a scale factor is achieved. In the calibration, we estimate the parameters of a geometric model based on n points whose 3D object coordinates $\mathbf{M}_i$ are known, whose 2D image coordinates $\mathbf{w}_i$ are measured, for $i = 1, ..., n$. Using the model we obtain the reprojected points $\mathbf{w}'_i = F(\mathbf{M}_i, \theta)$, *i.e.*, the inferred projections in the digital image computed from the calibration points $\mathbf{M}_i$ and a parameter vector θ. The calibration is performed in each image of the sequence by minimizing the objective function defined as the mean-square discrepancy between measured points $\mathbf{w}_i$ and inferred points $\mathbf{w}'_i$ [32]. Usually, the calibration problem is a non-linear optimization problem. In general, the minimization of the objective function has no closed-form solution. For this reason, it must be iteratively minimized starting with an initial guess θ_0 that can be obtained from nominal values or preliminary reference measurements.

ii) Uncalibrated Approach: The calibration is a very difficult task because the iterative estimation of the parameters is very sensible to the initial guess. In addition, the vibrations of the imaging system induce inaccuracies in the estimated parameters of the model, *i.e.*, the calibration is not stable and the computer vision system must be calibrated periodically (off-line) in order to avoid uncertainty. For this reason, we developed an approach based on the tracking of potential detects in two views [39,40] and in three views [40] using uncalibrated image sequences, in which it was not necessary to calibrate the imaging system. This new approaches track the potential defects based on a motion model estimated from the image sequence itself. Thus, we obtain a motion model by

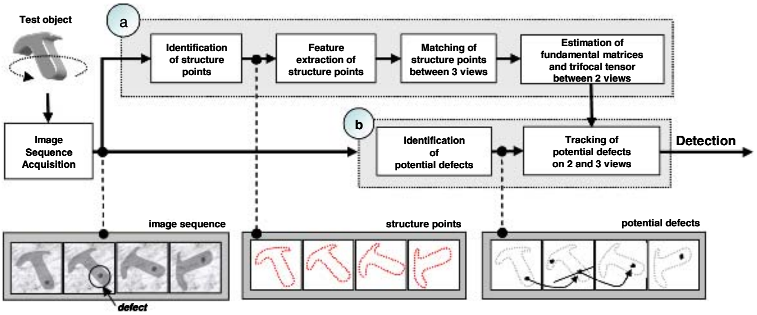

Test object
a
Identification of structure points
Feature extraction of structure points
Matching of structure points between 3 views
Estimation of fundamental matrices and trifocal tensor between 2 views
Image Sequence Acquisition
b
Identification of potential defects
Tracking of potential defects on 2 and 3 views
Detection
image sequence
structure points
potential defects
defect

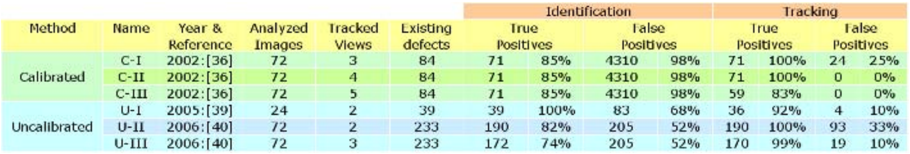
Fig. 2. Block diagram of the uncalibrated automated multiple view inspection: a) estimation of motion model, b) detection of defects [40]

matching structure points of the test object in the images as shown in Fig. 2. The structure points are matched using B-Spline curves and correlated curve sections of the structure (see details in [39] and in [40] respectively). Using RANSAC [32], the matched structure points are employed to estimated the bifocal and trifocal tensors required for the multiple view analysis. In this sense, we do not calibrate the image sequence, we only estimate the bifocal and trifocal tensors required for the tracking. The great disadvantage of this approach is the inherent difficulty in identification of the structure points (and thus the estimation of the motion model) from the test object itself, when the images of the test object do not significantly differ from each other in the sequence, *e.g.*, a glass or a bottle rotating around its vertical axis.

Once the system is calibrated (in the calibration approach) or the motion model is estimated (in the uncalibrated approach) the same algorithm is used to track the potential defects [41]. The tracking algorithm requires the bifocal and trifocal tensors [32] between the views. In the first approach the tensors are obtained from the projection matrices estimated after the calibration, whereas in the second approach the tensors are obtained using corresponding points of the test object in two and three views.

Table 1 summarizes the results obtained on real data using calibrated and uncalibrated approaches. We calculate the performance of the identification and the performance of the tracking separately. *True positives* are the number of defects correctly detected. The true positive percentage is calculated related to the number of the existing defects. *False positives* (or false alarms) correspond to the number of 'no-defects' misclassified as 'defects'. The false positive percentage is given related to the number of detected potential defects. We present three implementations of the calibrated approach. They perform the tracking in three, four and five views (cases C-I, C-II and C-III respectively). We observe that the number of false alarms in the identification is enormous. However, the results are perfect for four views (case C-II) where all defects are detected without any false alarms. The verification of the correspondence on three views flags too many false

Table 1. Performance of calibrated and uncalibrated approaches



						Identification				Tracking			
Method	Name	Year & Reference	Analyzed Images	Tracked Views	Existing defects	True Positives		False Positives		True Positives		False Positives	
Calibrated	C-I	2002:[36]	72	3	84	71	85%	4310	98%	71	100%	24	25%
	C-II	2002:[36]	72	4	84	71	85%	4310	98%	71	100%	0	0%
	C-III	2002:[36]	72	5	84	71	85%	4310	98%	59	83%	0	0%
Uncalibrated	U-I	2005:[39]	24	2	39	39	100%	83	68%	36	92%	4	10%
	U-II	2006:[40]	72	2	233	190	82%	205	52%	190	100%	93	33%
	U-III	2006:[40]	72	3	233	172	74%	205	52%	170	99%	19	10%

alarms. On the other hand, with 5 views we cannot ensure the segmentation of a defect in five views, for this reason some defects cannot be detected. We increase the performance in the segmentation in the uncalibrated approaches reducing the number of false alarms significantly. In case U-I, we perform the tracking in only two views using B-spline curves for the motion model. In case U-II and U-III, the tracking is done in two and three views respectively using correlated curve sections of the structure for the motion model. The results of case U-III are promising because all defects to be tracked, *i.e.*, defects that are present in three views, could be tracked, with only a few number of false alarms. We observe that the performance obtained in calibrated approach is higher, however the calibration is in many cases an excessively difficult and unstable task that can be avoided using an uncalibrated approach.

4 Conclusions

Automated visual inspection remains an open question. Many research directions have been exploited, some very different principles have been adopted and a wide variety of algorithms have been appeared in the literature of automated visual inspection. Although there are several approaches in the last 25 years that have been developed, automated visual inspection systems still suffer from i) detection accuracy, because there is a fundamental trade off between false alarms and miss detections; and ii) strong bottleneck derived from mechanical speed (required to place the test object in the desired positions) and from high computational cost (to determine whether the test object is defective or not). In this sense, Automated Multiple View Inspection offers a robust alternative method that uses redundant views to perform the inspection task. We believe that the method is opening up new possibilities in inspection field by taking into account the useful information about the correspondence between the different views of the test object. Two approaches were developed in the last years: the calibrated and the uncalibrated approaches. Both of them achieve very good performance. However, the calibration of the first approach is a very complicated task, and the identification of structure points in the second approach is inherently difficult when the images of the test object do not significantly differ from each other in the sequence. In order to avoid the mentioned problems, we are working on an on-line calibration of the multiple view system using a calibration object attached to the test object which is imaged in all views. Thus, the images have an enough number of points to calibrate the system.

Acknowledgments

This work was supported by FONDECYT – Chile under grant no. 1040210.

References

1. Malamas, E., Petrakis, E., Zervakis, M.: A survey on industrial vision systems, applications and tools. Image and Vision Computing **21** (2003) 171–188
2. Newman, T., Jain, A.: A survey of automated visual inspection. Computer Vision and Image Understanding **61** (1995) 231–262
3. Chin, R.: Automated visual inspection: 1981-1987. Computer Vision Graphics Image Process **41** (1988) 346–381
4. Chin, R.T., H.C.: Automated visual inspection: A survey. IEEE Trans. Pattern Analysis and Machine Intelligence **4** (1982) 557–573
5. Mery, D.: Automated radioscopic testing of aluminum casting. Materials Evaluation **64** (2006) 135–143
6. Castleman, K.: Digital image processing. Prentice-Hall, Englewood Cliffs, New Jersey (1996)
7. Purschke, M.: IQI-sensitivity and applications of flat panel detectors and X-ray image intensifiers - a comparison. Insight **44** (2002) 628–630
8. Davis, E.: Machine Vision. 3 edn. Morgan Kaufmann Publishers, Amsterdam (2005)
9. Trussell, H., Saber, E., Vrhel, M.: Color image processing. IEEE Signal Processing Magazine **22** (2005) 14–22
10. Nagy, J., Palmer, K., Perrone, L.: Iterative methods for image deblurring: A Matlab object-oriented approach. Numerical Algorithms **36** (2005) 73–93
11. Banham, M., Katsaggelos, A.: Digital image restoration. IEEE Signal Processing Magazine **14** (1997) 24–40
12. Zhang, Y.J.: An Overview of Image and Video Segmentation in the Last 40 Years. In: Advances in Image and Video Segmentation, Zhang, Y.-J. (Ed.). IRM Press, Idea Group Inc. Hershey (2006) 1–15
13. Cheng, H., Jiang, X., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. Pattern Recognition **34** (2001) 2259–2281
14. Koschan, A., Abidi, M.: Detection and classification of edges in color images: A review of vector-valued techniques. IEEE Signal Processing Magazine **22** (2005) 64–73
15. Mery, D.: Crossing line profile: a new approach to detecting defects in aluminium castings. Lecture Notes in Computer Science **2749** (2003) 725–732
16. Mery, D., Berti, M.: Automatic detection of welding defects using texture features. Insight **45** (2003) 676–681
17. Webb, A.: Statistical Pattern Recognition. 2 edn. John Wiley and Sons Ltd., New Jersey (2005)
18. Jain, A., Duin, R., Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Analysis and Machine Intelligence **22** (2000) 4–37
19. Somol, P., Pudil, P., Kittler, J.: Fast branch and bound algorithms for optimal feature selection. IEEE Trans. on Pattern Analysis and Machine Intelligence **26** (2004) 900–912
20. Mery, D., da Silva, R., Caloba, L., Rebello, J.: Pattern recognition in the automatic inspection of aluminium castings. Insight **45** (2003) 475–483

21. Carvajal, K., Chacón, M., Mery, D., Acuña, G.: Neural network method for failure detection with skewed class distribution. Insight **46** (2004) 399–402
22. Kay, S.: Fundamentals of Statistical Signal Processing: Detection Theory. Prentice Hall Signal, Processing Series, Volume 2, New Jersey (1998)
23. Provost, F., Fawcett, T.: Robust classification for imprecise environments. Machine Learning Journal **42** (2001) 203–231
24. Manolakis, D., Shaw, G.: Detection algorithms for hyperspectral imaging applications. IEEE Signal Processing Magazine **19** (2002) 29–43
25. Mery, D., Hahn, D., Hitschfeld, N.: Simulation of defects in aluminium castings using cad models of flaws and real X-ray images. Insight **47** (2005) 618–624
26. Raina, R., Shen, Y., Ng, A.Y., McCallum, A.: Classification with hybrid generative/discriminative models. In: Advances in neural information processing systems 16, S. Thrun, L. Saul , and B. Schlkopf (Eds.). MIT Press, In Cambridge, MA (2003)
27. Theodoris, S., Koutroumbas, K.: Pattern Recognition. 2 edn. Academic Press, Elsevier, Amsterdam (2003)
28. Bishop, C.: Neural Network for Pattern Recognition. Oxford University Press Inc., New York (1997)
29. Shawe-Taylo, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, UK (2005)
30. Kuncheva, L.: A theoretical study on six classifier fusion strategies. IEEE Trans. on Pattern Analysis and Machine Learning **24** (2002) 281–286
31. Mery, D., Chacón, M., Muñoz, L., Gonzalez, L.: Automated inspection of aluminium castings using fusion strategies. Materials Evaluation **63** (2005) 148–153
32. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, UK (2000)
33. Mery, D.: Exploiting multiple view geometry in X-ray testing: Part I: Theory. Materials Evaluation **61** (2003) 1226–1233
34. Mery, D.: Exploiting multiple view geometry in X-ray testing: Part II: Applications. Materials Evaluation **61** (2003) 1311–1314
35. Mitchell, T.: Machine Learning. McGraw-Hill, Boston (1997)
36. Mery, D., Filbert, D.: Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence. IEEE Trans. Robotics and Automation **18** (2002) 890–901
37. Mery, D.: Explicit geometric model of a radioscopic imaging system. NDT & E International **36** (2003) 587–599
38. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. on Pattern Analysis and Machine Intelligence **22** (2000) 1330–1334
39. Mery, D., Carrasco, M.: Automated multiple view inspection based on uncalibrated image sequences. Lecture Notes in Computer Science **3540** (2005) 1238–1247
40. Carrasco, M., Mery, D.: Automated visual inspection using trifocal analysis in an uncalibrated sequence of images. Materials Evaluation **64** (2006) In Press
41. Mery, D., Ochoa, F., Vidal, R.: Tracking of points in a calibrated and noisy image sequence. Lecture Notes in Computer Science **3211** (2004) 647–654

Target Tracking and Positioning on Video Sequence from a Moving Video Camera

Chi-Farn Chen and Min-Hsin Chen

Center for Space and Remote Sensing Research
National Central University
Jhongli, Taiwan

Abstract. Video images have long been used in tracking a target. Most of these studies have focused on tracking the moving target with a motionless video camera. Yet, relatively few studies have been conducted on target tracking from a moving video camera. In addition, the tracking of a moving target from one single video camera is inherently unable to provide the positioning information of the target in real space. This study proposes a three-stage approach with the concept of segmentation and photogrammetry to track and position the target on the video sequence acquired from a moving video camera. In the first stage, both the color segmentation technique and ACM (Active Contour Model) are used to separate the target from the background. In the second stage, an area-based matching approach is employed to track the feature points of the target on the video sequence. In the final stage, by integrating the video camera, GPS, digital compass and tracking results, the ground position of the target can be calculated using the photogrammetric techniques.

Keywords: Object Tracking, Object Positioning, Moving Video camera.

1 Introduction

Digital video cameras have recently become a popular monitoring tool, where it can be used for intelligent vision applications, as it is smaller, lighter and easier to use than traditional ones. Over the past years, a considerable amount of studies have been made on video sequence analysis and vision applications, such as security surveillance [1], traffic monitoring [2], and human movement analysis [3][4]. In these studies, numerous algorithms have been developed in tracking a moving target with the motionless video camera [5][6][7]. Since these algorithms need a relatively stationary background to track the moving target successfully, the video camera is generally kept motionless. For example, the conventional background suppression [8][9] and frame differencing [10][11] methods are able to effectively track the moving target via a motionless video camera. Moreover, the tracking information of these studies is generally presented on the image plane instead of transforming the track into the position information in real space. Nevertheless, very few attempts have been conducted regarding the issue of tracking and positioning the target on video sequence acquired from a moving video camera. In fact, many monitoring applications require not only the track information of the target, but also its position

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 523 – 533, 2006.

data in real space. One of the major applications is the position data (e.g. the coordinates) of the target can be integrated with GIS (Geographic Information Systems) for further spatial analysis. It is well known that the coordinates of the target in real space are impossible to be obtained solely from a camera, when its orientation parameters are unknown [12]. The orientation parameters can be mathematically determined if the target is recorded by at least two cameras or by a moving video camera. In this study, we propose a three-stage approach, utilizing the concept of segmentation and photogrammetry to track and position the target on the video sequence acquired from a moving video camera. The proposed approach consists of three stages: (1) target segmentation, (2) target tracking and (3) target positioning. In the first stage, although the appearances of the background on each image frame will vary from time to time due to the moving motion of the video camera, the colors of the target's major features will mostly be preserved on the adjacent image frames. Thus, the color segmentation technique is used to separate the target from the background. In addition, in order to preserve the target after the segmentation process, the Active Contour Model (ACM) is used to maintain the target's proper shape. In the second stage, an area-based matching approach is employed to track the feature points of the target on the video sequence. In the final stage, by integrating the video camera, GPS, digital compass and tracking results in each frame; the ground position of the target can be calculated using the collinearity condition equations and the least squares adjustments.

The organization of this paper is as follows. Section 2 introduces the hardware used in the proposed method. Section 3 describes the methodology that includes the target extraction, the target tracking and the target positioning. Three experiments are designed to test the proposed approach and the results will be shown in Section 4. The conclusions will be addressed in Section 5.

2 The Hardware System

Basically, both the position and orientations of a camera, which is called the exterior orientations in photogrammetry, are necessary to calculate the position of a target in real space. In order to record the video images and the exterior orientations of the camera synchronously, the GPS, digital compass and tilt meter with a digital video camera, were integrated through an encoded-decoded hardware device in this study. The type of digital video camera used in recording the data is the SONY PC115, whose super steady shot function can provide better image quality during the actual movements. The accuracy of the tilt meter is +/- 1° with a range of +/- 20° from the horizon. The accuracy of the digital compass is about +/- 3°, and the GPS accuracy is roughly in 10~15 meters. In addition, the data update rate of the entire system is limited to 1 Hz, because the GPS receiver can only provide the data at 1 Hz. After the recording procedure is accomplished, the exterior orientations of the video camera and the corresponding video sequence can be captured in a digital format synchronously by employment of the decoded device and Microsoft DirectX Software Development Kit.

3 Methodologies

The methodologies used in the proposed approach can be divided into three main stages. The first stage uses both color-based and contour-based segmentation techniques to extract the target. The second stage implements the tracking task by matching the feature points of the target. In the third stage, the ground position of the target can be calculated by integrating the camera's orientations, target tracking results and photogrammetric techniques. A systematic flowchart of the proposed approach is illustrated in Fig1. The following sections will describe the physical mechanisms of each stage in greater detail.

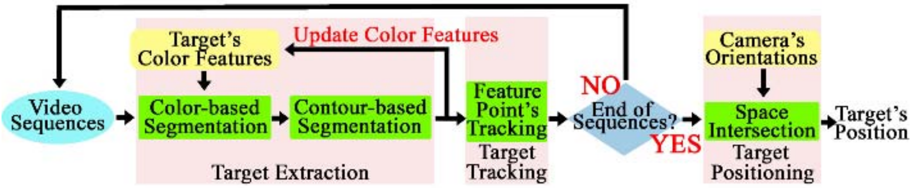


Fig. 1. Flowchart of the proposed approach

3.1 Target Extraction

In this segment, details of the target extraction method will be presented. As the image background may vary continuously, due to the video camera's constant movements, the commonly used image differentiating method can not be properly employed to extract the target. Thus, we use a two-step segmentation procedure to constantly track the target, image by image, from the video sequence. In the first step, the major color features of the target is extracted, and used to represent the target. In the second step, the ACM is applied to preserve the target's appropriate shape.

3.1.1 Color-Based Segmentation

During the human eye's initial perception of the object, similar colors are always grouped together for further analysis. Based on this assumption, the target can be represented as features with characteristic colors [13]. However, because of the motion of the video camera, the colors of the background may change and generate a color confusion associated with the target, during the target extraction. In this study, the preferred target is manually selected by marking a rough polygon line in the first image. Afterward, a buffer region surrounding the target is automatically generated to represent the background. In order to avoid confusion of the color features, similar color features between the target and the background have to be eliminated from the target during the target extraction. First, an unsupervised classification (K-mean classifier) method is used to extract the different color classes of the target and the background, respectively. Second, based on the spectral distance between the target and the background classes, a comparison is made to eliminate the classes with a shorter spectral distance that signifies the color similarity. By doing so, it will avoid the color confusion between the target and background. Both Fig. 2 (a) and 2(b) depict the results of the color-based segmentation. Fig. 2(a) indicates that the similar color classes of both the target and background can be eliminated from the target, due to their color similarities. Fig. 2(b) shows the rest of the color features on the target. As Fig. 2(b) illustrates, the target's shape is quite different from the original one. In order to maintain the target's proper shape, the Active Contour Model (ACM), which is based on an energy-minimizing contour segmentation algorithm, is used to describe the target's resulting shape. The detailed procedures of the ACM will be seen in the next sub-section.

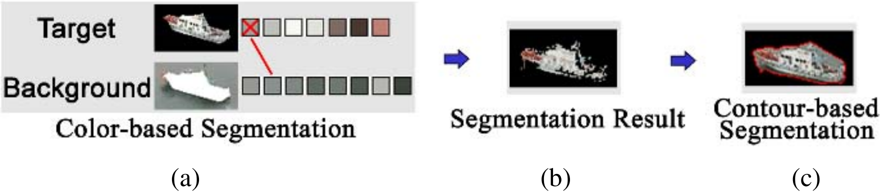


Fig. 2. Sketch graphic of the target extraction: (a) Color features extraction of both the target and background. (b) Results of the color-based segmentation. (c) Results of Contour-based segmentation.

3.1.2 Contour-Based Segmentation

The Active Contour Model (ACM) [14] is a kind of parametric curve presentation, which is defined within a curve domain. The curves of the ACM can move under the influence of internal forces, caused by the initial curve, and external forces supplied by image data. The ACM is widely used in image processing applications, such as edge detection, segmentation and particularly in the location of object boundaries [15]. The ACM transfers the boundary detection problem in the image domain to the energy-minimizing problem in the curve domain. The traditional energy functions in the ACM are defined as follows:

$$E_{Snake} = \sum_{i=1}^{N} (E_{int}(s_i) + E_{ext}(s_i)) \tag{1}$$

$$E_{\text{int}}(S_i) = \alpha |S_i - S_{i-1}|^2 + \beta |S_{i-1} - 2S_i + S_{i+1}|^2 \tag{2}$$

$$E_{ext}(S_i) = -|\nabla f(S_i)| \tag{3}$$

Where

E_{Snake} : ACM energy of the contour

S_i : ith Seed pixel

N : Number of seed pixel

$E_{\text{int}}(S_i)$: Internal energy at seed pixel S_i

$E_{ext}(S_i)$: External energy at seed pixel S_i

α、β : The weighting functions are defined to control the importance of the elastic and bending terms

Traditional ACM has two problems: initialization and convergence in concave regions. The initialization problem means that the initial contour has to be close to the object, because the potential force of traditional ACM is generally small. In addition, due to the fact that the ACM has no extra pressure force on the concave region; the contour often goes across the boundary concave. In order to solve these problems, an additional external force, the Gradient Vector Flow (GVF), [16] was developed to improve the result of the ACM. The GVF is a diffusive field, which is computed from gradient vectors of a gray-level map derived from the image. When the point in the field is near the object's boundary, the GVF field will move toward the boundary. Moreover, the GVF field will change smoothly in the homogenous regions of the image. Therefore, the GVF is not only capable of providing a larger buffer in the initial contour, it will also converge in the concave region as well.

Since the target object may have irregular shapes, the value of elastic parameter α and the bending parameter β used in this study are given as 0.7 and 0.015. The resulting image of the color-based segmentation can be transferred to a binary image (Fig. 3a, 3b). The GVF can then be produced from the binary image. Fig. 3c indicates the intensity of the GVF, where the pixels with higher brightness levels have higher intensities of the GVF. In this study, the boundary of each search rectangle is considered an initial contour for the ACM. Fig. 3d, 3e and 3f show the process of the ACM. The red contour represents the contour of the ACM, as the contour is evidently seen approaching the target pixels, when the iterations increase. Eventually, by using the GVF Active Contour Model, the target object can be automatically and completely segmented from the video images.

Since the color features of background may change due to the motion of the video camera, it is necessary to update the color features of both the target and the background to avoid the color confusion in the following sequence.

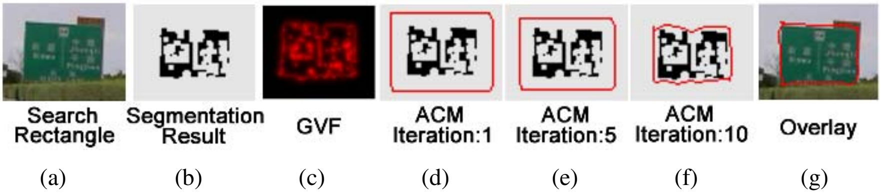


Fig. 3. Sketch graphic of the contour-based segmentation. (a) Image of search rectangle. (b) Binary image of the target segmentation. (c) GVF of binary image. (d) Contour of ACM in iteration 1. (e) Contour of ACM in iteration 5. (f) Contour of ACM in iteration 10. (g) Overlay of the final ACM contour on the original image.

3.2 Target Tracking

After the execution of both the color-based and contour-based segmentation on the first image, a color-exclusive target will be generated. The polygon line of the target will be moved to the next image, and will be used to repeat both the color and contour segmentation on the image. As a result, a series of color-exclusive target images will be produced from the video sequence. In general, the procedures will complete the target image tracking. Since the target positioning requires the point coordinates to calculate the position, this study uses the following procedures to pin down the target image to the point coordinates.

In this approach, the TDGO [17] operator is used to extract the most obvious feature point of the target object. Afterwards, an area-based matching algorithm is applied to track the feature point. Since the target object has already been extracted during the previous stage, it is helpful to reduce the matching error probability, and increase the efficiency of the matching procedure. This study selects the Mean Square Error (MSE) to be the objective function of an area-based matching. The MSE measures the error magnitude via a two blocks' comparison. A lower error means a higher probability of the two blocks being similar. The definition of the MSE is shown in Eq. (4). Let B_{pq} be the candidate block with position (p,q) in $N+1_{th}$ frame, and T_{rs} as the target block with position (r,s) in N_{th} frame. The Mean Square Error of the two blocks is given as:

$$MSE(B_{pq},T_{rs}) = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(B_{pq}^{ij} - T_{rs}^{ij} \right)^2 \tag{4}$$

Where:

M, N : The dimensions of the block

i, j : Pointer of pixel in blocks

3.3 Target Positioning

As stated in section 2, the exterior orientations of the camera can be obtained from the hardware system. Furthermore, the target segmentation and feature point tracking can help locate the image coordinates of the target. Consequently, for each frame with exterior orientation data, it will generate three equations. Based on the concept of the space intersection in photogrammetry, the equation (Eq. 5) can be derived from the relationships between the camera's position *P*, scaling factor *S*, unit vector of observation *U* and the target's location *T* in a local geodetic system *(E, N, H)*. The geodetic system used in this research is the TWD67, which is widely used in Taiwan (unit is depicted in meters). A rotation matrix is required to transfer a camera coordinate system to a geodetic one. (Eq. 6,7).

$$T = P^i + S^i \cdot U^i \tag{5}$$

$$U^i = R_{FB}^i \cdot Obs^i / \left| Obs^i \right| \tag{6}$$

$$R_{FB}^i = \begin{bmatrix} c(\phi^i)c(\kappa^i) & s(\omega^i)s(\phi^i)c(\kappa^i) + c(\omega^i)s(\kappa^i) & -c(\omega^i)s(\phi^i)c(\kappa^i) + s(\omega^i)s(\kappa^i) \\ -c(\phi^i)s(\kappa^i) & -s(\omega^i)s(\phi^i)s(\kappa^i) + c(\omega^i)c(\kappa^i) & c(\omega^i)s(\phi^i)s(\kappa^i) + s(\omega^i)c(\kappa^i) \\ s(\phi^i) & -s(\omega^i)c(\phi^i) & c(\omega^i)c(\phi^i) \end{bmatrix} \tag{7}$$

Where:

T : Position vector of Target $[T_E \ T_N \ T_H]^T$

P^i : Position vector of frame i. $[P_E^i \ P_N^i \ P_H^i]^T$

S^i : Scaling factor of frame i

U^i :Unit vector pointing from the camera to the target of frame i. $[U_E^i \ U_N^i \ U_H^i]^T$

R_{FB}^i : Rotation matrix of frame i, the shorthand ***c*** for cos () and ***s*** for sin (), ω 、ϕ 、κ individually represent the pitch 、roll and heading of camera

Obs^i : Observation vector $[x^i \ f \ y^i]^T$, x^i and y^i are the image coordinates of target in frame i.

If the duration of the whole tracking procedure is in **n** seconds, there will be **n** images with both exterior and interior orientation data. This is attributed by the fact that the system's data update rate is 1Hz. Moreover, because all the observation vectors in this study are 3-dimension vectors (E, N and H), the **n** observation vectors can construct **3n** equations. Under the condition of **n** observation vectors, the number of unknowns is **n+3**, which consists of three ground coordinates of target $[T_E \ T_N \ T_H]^T$, and **n** scaling factors for each observation vector (Eq. 8). Since the observation vectors have already been normalized to a unit vector, the scaling factor can accordingly be considered as the distance between the target and the camera. It is clear that when **n** is greater than 2, the number of equations will be greater than the number of unknowns. Therefore, by using the least squares adjustment, the most approximate position of the target can be estimated.

$$\begin{bmatrix} 1 & 0 & 0 & -U_E^1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & -U_N^1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -U_H^1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & -U_E^2 & \cdots & 0 \\ 0 & 1 & 0 & 0 & -U_N^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 & 0 & \cdots & -U_H^n \end{bmatrix}_{[3n][n+3]} \bullet \begin{bmatrix} T_E \\ T_N \\ T_H \\ S_1 \\ S_2 \\ \vdots \\ S_n \end{bmatrix}_{[n+3][1]} = \begin{bmatrix} P_E^1 \\ P_N^1 \\ P_H^1 \\ P_E^2 \\ P_N^2 \\ \vdots \\ P_H^n \end{bmatrix}_{[3n][1]} \quad (8)$$

4 Experimental Results and Discussions

Three experiments are conducted by the video system described in section 2. In the first experiment, the video system is mounted on a helicopter that flies around the coast to target a stationary ship. In both the second and third experiment, the video system is placed on a car traveling on a highway targeting a signboard and the Taipei 101 building, respectively. The video sequence of the three experiments is recorded at a frame rate of 10 fps, within a 320 by 240 image size. The video sequence segments and the results of the target extraction and tracking of the three experiments are presented, respectively, in Fig.4, Fig.5 and Fig.6. The results of the target position can be seen in Table1. In addition, the ortho aerial photos are also used in conjunction with Fig.4, Fig.5 and Fig.6 for map referencing. The orange triangles represent the GPS positions of the moving video camera, the red circle dot indicates the actual coordinates of the target, and the blue square box denotes the estimated coordinates of the target, derived from the proposed method. Throughout the remaining part of this paper, the three experiments will be referred to as the "Boat experiment", "Signboard experiment" and "Taipei 101 experiment" respectively.

For the Boat experiment, the helicopter flies roughly at 50 km/h at a height of about 280 m, and the video recording time is 30 seconds. A base-height ratio can be generated from the data, where the base represents the total displacement of the moving camera, while the height depicts the average distance between the moving camera and target. The Boat experiment will produce a base-height ratio approximately equal to one (Table 1), which indicates that an excellent space geometry can be created. This offers a conducive condition for position estimations. As the helicopter flies around the boat in circles during the experiment, various shapes and sizes of the boat images, and unpredictable color variations of the background images are produced (Fig.4). It is obvious that these random variations of the target and background images generate difficulties for target extraction and tracking. A segment of the video sequence and the results of both the target extraction and tracking are shown in Fig.4. A close inspection of the "Target Extraction" in Fig.4 indicates that the boat can be effectively extracted from the background, and the shape of the boat can be well preserved (The boat is encircled by the red contour). The target tracking results are presented in "Target Tracking" in Fig.4, where the sign of the cross indicates the position of the feature's point on the target image. It is seen that every target image is correctly spotted with the cross. In order to perform the accuracy check for the estimated coordinates of the boat's feature point, which are derived from the target positioning, a ground survey from the coastland using two electronic altazimuths via a space intersection technique is used to measure the actual coordinates of the ship. The position comparison indicates a position discrepancy of approximately 12.5 meters (Table 1). If the point position of the ship can be considered as a part of the ship body (the ship is about 50 meters in length), the position discrepancy may be practically ignored in the position comparison. It should be reasonable to conclude that the Boat experiment for boat positioning is considered as the successful one.

During the Signboard experiment, the car travels at a speed of about 70km/h. The average filming distance is about 84m, and the video recording time is 6 seconds. With a base-height ratio approximately equal to one (Table 1), it may create an excellent space geometry and advantageous conditions for position estimations. As the experiment is designed to drive the car on the highway, and record the signboard from the window, a series of gradually bigger images of the signboard are recorded. A segment of the video sequence, along with the results of both the target extraction and tracking are shown in Fig.5. The "Target Extraction" and "Target Tracking" of Fig.5 clearly reveal that the proposed method successfully extracts and tracks the signboard. The actual coordinates of the signboard are measured by GPS, and used to compare with the estimated coordinates. The accuracy check shows a position discrepancy of about 2.7 meters (Table 1). The small discrepancy can basically be attributed to the success of the target extraction and tracking, as well as the excellent space geometry generated from a good base-height ratio (Table 1).

For the Taipei 101 experiment, the car travels at a speed of about 55km/h. The average filming distance is about 3380m, and the video recording time is 20 seconds. Since there are some visual blockages in the line of sight between the camera and the

Taipei 101 building, the recording baseline can only reach around 290m. Since the video is recorded with the long observation distance and a relatively short baseline, the base-height ratio in this case is approximately equal to 0.1 (Table 1). With such a low base-height ratio, it represents the poor space geometry and the weak conditions for position estimations. A portion of the video sequence and the results of both the target extraction and tracking are shown in Fig.6. The "Target Extraction" and "Target Tracking" of Fig.6 visibly show that the proposed method successfully extracts and tracks the Taipei 101 skyscraper. The performance of the accuracy check is carried out by comparing the estimated coordinates with the actual coordinates measured from the ortho aerial photo. The accuracy check indicates that a position discrepancy of about 98.4 meters can be found (Table 1). In spite of the success of the target extraction and tracking, the poor space geometry is apparently responsible for the relatively large discrepancy.

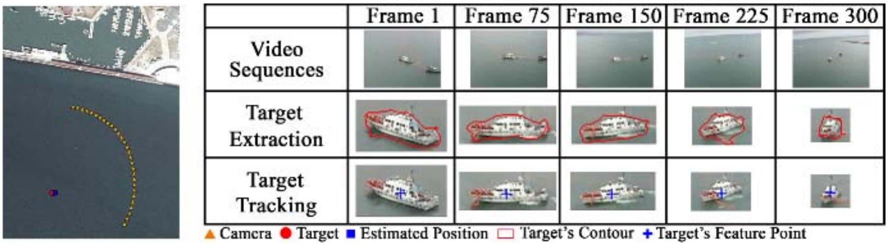


Fig. 4. Test results of Boat experiment

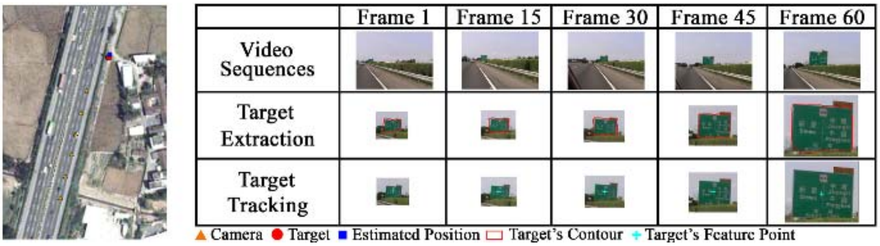


Fig. 5. Test results of Signboard experiment

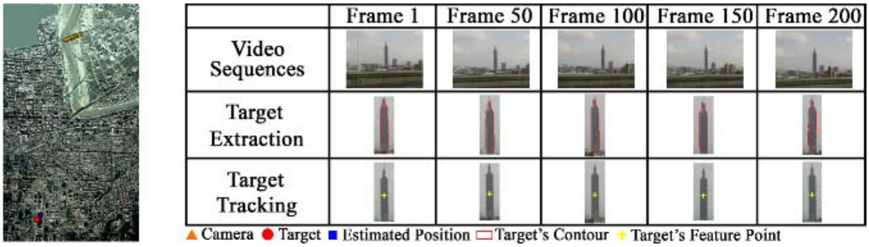


Fig. 6. Test results of Taipei 101

Table 1. Position discrepancy between the proposed method and the ground survey

Experiment	Actual Coordinates E, N (m)	Estimated Coordinates E, N (m)	Base (m) – Height (m) Ratio	Discrepancy in E, N (m)	Total Discrepancy (m)
Boat	290581, 2785809	290587.9, 2785819.5	405/280=1.446	6.9, 10.4	12.5
Signboard	268869, 2760042	268870.2, 2760045	98/84=1.16	1.2, 2.4	2.7
Taipei 101	306120, 2769925	306179.3, 2770003.8	293/3381=0.087	59.3, 78.5	98.4

5 Conclusions

This paper introduces a three-stage approach to track and position the target from a moving video camera. In the first stage, both the color-based and contour-based segmentation techniques are used to separate the target from the background. In the second stage, in order to pin down the target images to the point coordinates, an area-based matching approach is employed to track the feature points of the target on the video sequence. In the final stage, the ground position of the target can be calculated via integrating the camera's orientations, tracking results of the target and techniques found in photogrammetry. Three experiments are designed to test the possibility and reliability of the proposed method. The results indicate that the proposed approach can successfully extract and track the target on a moving camera. The accuracy checks of the target positioning show that the position discrepancies between the estimated coordinates and the actual coordinates may be essentially ignored if the experiments can obtain a good space geometry. In contrast, the occurrence of the relatively large position discrepancy can be basically attributed to the limited precision of the recording devices, as well as the poor space geometry during the image recording.

References

1. Meyer, M.; Ohmacht, T.; Bosch, R.; Hotter, M.: Video surveillance applications using multiple views of a scene, Aerospace and Electronic Systems Magazine, IEEE Volume 14, Issue 3. (1999) 13 - 18
2. Atiya, A.F.; Aly, M.A.; Parlos, A.G.: Sparse basis selection: new results and application to adaptive prediction of video source traffic, Neural Networks, IEEE Transactions on Volume 16, Issue 5. (2005) 1136 - 1146
3. Xangdong Xie; Sudhakar, R.; Hanqi Zhuang: Real-time eye feature tracking from a video image sequence using Kalman filter, Systems, Man and Cybernetics, IEEE Transactions on Volume 25, Issue 12. (1995) 1568 - 1577
4. Veeraraghavan A; Roy-Chowdhury, A.K.: Chellappa, R.: Matching shape sequences in video with applications in human movement analysis, Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 27, Issue 12. (2005) 1896 – 1909
5. Grinias and G. Tziritas.: A semi-automatic seeded region growing algorithm for video object localization and tracking. Signal Processing: Image Communication. (2001) 977-986

6. Cucchiara, R.; Prati, A.; Vezzani, R.: Object segmentation in videos from moving camera with MRFs on color and motion features, Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on Volume 1. (2003) I-405 - I-410
7. Yi Liu; Zheng, Y.F.: Video object segmentation and tracking using /spl psi/-learning classification, Circuits and Systems for Video Technology, IEEE Transactions on Volume 15, Issue 7. (2005) 885 - 899
8. Horton, R.D.: A target cueing and tracking system (TCATS) for smart video processing, Aerospace and Electronic Systems Magazine, IEEE Volume 6, Issue 3. (1991) 8 – 13
9. Cavallaro, A.; Steiger, O.; Ebrahimi, T.: Tracking video objects in cluttered background, Circuits and Systems for Video Technology, IEEE Transactions on Volume 15, Issue 4. (2005) 575 – 584
10. Wei Niu; Jiao Long; Dan Han; Yuan-Fang Wang, Human activity detection and recognition for video surveillance, Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on Volume 1. (2004) 719 - 722
11. Dailey, D.J.; Li, L., An algorithm to estimate vehicle speed using un-calibrated cameras, Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on 5-8 Oct. (1999) 441 – 446
12. Paul R. Wolf and Bon A. Dewitt, Elements of Photogrammetry with Application in GIS, McGRAW-Hill, United States. (2000)
13. Harper, P.; Reilly, R.B.: Color based video segmentation using level sets, Image Processing, 2000. Proceedings. 2000 International Conference on Volume 3, 10-13 Sept. (2000) 480 – 483
14. Kass, M. A. and D. Terzopoulos, Snakes: Active contour models, in Proc. 1st International Conference Computer Vision, London. (1987) 259-268.
15. Lin Yang; Meer, P.; Foran, D.J.: Unsupervised segmentation based on robust estimation and color active contour models, Information Technology in Biomedicine, IEEE Transactions on Volume 9, Issue 3. (2005) 475 – 486
16. Xu, C. and J. L. Prince, Generalized Gradient Vector Flow External Forces for Active Contours, Signal Processing─An International Journal, vol. 71, no. 2. (1998) 131-139
17. Lue, Y., Interest Operator and Fast Implementation, International Archives of Photogrammetry and Remote Sensing, 27(Ⅱ). (1988) 491-500

Radiometrically-Compensated Projection onto Non-Lambertian Surface Using Multiple Overlapping Projectors

Hanhoon Park, Moon-Hyun Lee, Byung-Kuk Seo, Hong-Chang Shin, and Jong-Il Park

Division of Electrical and Computer Engineering, Hanyang University, Seoul, Korea
{hanuni, fly4moon, nwseoweb, sesias}@mr.hanyang.ac.kr,
jipark@hanyang.ac.kr

Abstract. Existing radiometric compensation methods are based on the assumption that the projection surface (screen) is Lambertian, i.e. there exists no specular reflection. Thus the methods cannot be applied to non-Lambertian surfaces which are ubiquitous in our everyday environment. In this paper, we try to faithfully display an image onto non-Lambertian surfaces using multiple overlapping projectors and cameras. The projectors which are separated at a distance would hardly produce specular reflection at the same time at the same point of the projection surface. Therefore, we can reasonably assume that there is at least one *diffuse projector* which does not generate specular reflection in a region of projection surface at a camera viewpoint. From the perspective of the diffuse projector-camera pair, the region of projection surface looks like Lambertian and the existing radiometric compensation methods could be employed for compensating the radiometric distortion of the region. Experimental results are given to show the validity of our method.

Keywords: Radiometric compensation, projection-based augmented reality, multiple overlapping projectors, non-Lambertian surface.

1 Introduction

Recently, projection-based display systems have gained attention and been applied to many applications such as generating immersive 3-D virtual environment [11] or changing the color and texture of real objects [12]. Projection made it possible to use 3-D real and large objects as displays [16] and freed from discomforts inherent to wearing a device such as HMD. In the projection-based display systems, 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 (distorted) by the surface color. The radiometric compensation is a technique that makes the color of projection look unchanged by adjusting the color of the projection input in advance when the projection surface has colorful texture [3, 9, 1]. In the projection-based display systems, video projectors and cameras are usually used for projecting an image and observing the projected image respectively.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 534 – 544, 2006.

The previous radiometric compensation methods are based on the assumption that the projection surface is Lambertian, i.e. there exists no specular reflection. Actually, it may be an unconvincing assumption because the projection surfaces are non-Lambertian in our everyday environment. However, it is not easy to modify the previous radiometric compensation methods themselves such that they can cope with specular reflection.

Recently, Park et al. have proposed a method for projecting an image onto non-Lambertian surfaces while avoiding specular reflection using multiple overlapping projectors [7, 8]. Their method enabled us to catch diffuse reflection selectively at a camera viewpoint, exclusive of specular reflection. This ability inspired us to utilize multiple overlapping projectors for radiometrically-compensated projection onto non-Lambertian surfaces.

In this paper, our initial system using two projectors and a single camera is presented. However, there is no doubt that the method can be easily extended to the case of M projectors and N cameras. In the experiments, it is assumed that the projection surfaces are smooth and thus piece-wise planar. This assumption allows the geometric relationship between projectors, camera, and projection surface to be defined by *homographies* [9, 15, 8].

2 Radiometric Compensation of Non-Lambertian Surfaces Using Multiple Overlapping Projectors

Multiple overlapping projectors have been successfully used for avoiding specular reflection in a direct-projected augmented reality system [7, 8] or shadow due to projector light as a part of an intelligent presentation system [2]. In this paper, we utilize multiple overlapping projectors for enabling radiometric compensated projection onto non-Lambertian surfaces. The concept is not much different from the previous methods [7, 8, 2]. If multiple projectors are separated at a distance, there would exist at least one projector which does not generate specular reflection in a region of projection surface at a camera viewpoint. We call the projector *diffuse projector* in the region at the viewpoint. In the other regions and at the other camera viewpoint, there would exist another diffuse projector. Therefore, at an arbitrary camera viewpoint, the data for radiometric compensation in a certain region of the projection surface could be computed from the diffuse projector of the region at the viewpoint and camera pair.

After radiometric compensation, at each point of the projection surface, light of the projectors which are not *diffuse projector* is blocked and light (radiometrically compensated) of the diffuse projectors is boosted to make the brightness of the final projection unchanged [7, 8].

Notice that it should be known which projector generates specular reflection or not with respect to each point of the projection surface. In this paper, a surface normal based measure is used for this [8].

Note that the performance of the proposed method depends on the reflection property of the projection surface material. If a material cannot have sufficient diffuse reflection, the proposed method may not be useful for the material. If a material has excessive specular reflection, the results may look still distorted even after radiometric compensation. Actually, the proposed method would gain an advantage over existing radiometric compensation methods when the projection surface material is suitably specular and also has sufficient diffuse reflection. In this paper, we measure a simplified isotropic BRDF to examine the reflection property of several typical materials and heuristically adopt a material most suitable for verifying the validity of the proposed method.

3 Geometric Registration

In this section, it is explained how multiple projectors are overlapped. Once the projectors and the camera are calibrated using a variant of the Zhang's calibration method [9, 8], the projection surface geometry is recovered first using a simple binary-code method based on temporal coding [13]. A set of patterns are successively projected onto the surface and imaged at a camera. The codeword for a given pixel is usually formed by the sequence of illumination value for that pixel across the projected patterns. The correspondences between the projector input images and its corresponding camera image are found based on the codeword [13]. Finally, their 3D position is computed by applying triangulation [4] to the corresponding pixels of the projector input image and the camera image. The surface is triangularly represented based on the recovered points on the surface.

Because the surface is represented piece-wise planar (by triangle meshes), the geometric relationship between projectors, camera, and the projection surfaces can be defined by homographies [9, 15, 8]. The Discrete Linear Transform (DLT) algorithm can be used to calculate homographies between the projectors and camera via the triangle meshes. More sophisticated algorithms can be used for obtaining a reliable solution [4].

Once the homographies between the projector input images and the camera images are estimated, the projector input images are prewarped to be overlapped. Let H_{P_i-C} be the homography that takes the *i-th* projector coordinates to the camera coordinates. Then, the projector input images are pre-warped by the following transformation.

$$\begin{aligned} &x_i'' = x_i'/z_i', \quad y_i'' = y_i'/z_i', \\ &[x_i' \quad y_i' \quad z_i']^T = H_{P_i-C}^{-1}[x \quad y \quad 1]^T. \end{aligned} \tag{1}$$

Here, *x* and *y* are the original projector input image coordinates, x_i'' and y_i'' are the pre-warped projector input image coordinates. Then, the coordinates of the pre-warped projector input images in the camera coordinate system (u_i'', v_i'') becomes same (overlapped) as each other as follows.

$$\begin{aligned} &u_i'' = u_i'/w_i', \quad v_i'' = v_i'/w_i', \\ &[u_i' \quad v_i' \quad w_i']^T = (1/z_i')H_{P_i-C}[x_i'' \quad y_i'' \quad 1]^T \\ &\qquad = H_{P_i-C}H_{P_i-C}^{-1}[x \quad y \quad 1]^T = [x \quad y \quad 1]^T. \end{aligned} \tag{2}$$

4 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. Radiometric compensation is a technique that makes the color of projection look unchanged by adjusting the color of the projector input image in advance when the projection surface has colorful texture [3, 9].

In this section, it is assumed that the projection surface is Lambertian. Letting *I* be the projector input image, the projected image I_p is acquired by projector response function *f* as

$$I_P = f(I)\,. \tag{3}$$

The projected image is observed by a camera. The radiometric model of the pipeline from the projection color to the camera input radiance color *R* is defined as

$$R = VI_P + F \quad \text{where } V\text{: color mixing matrix, } F\text{: ambient light.} \tag{4}$$

The measured camera image *C* is acquired by camera response function *g* as

$$C = g(R)\,. \tag{5}$$

Once the unknown parameters are estimated, the projector input image is compensated in advance to obtain the desired camera image $C_{desired}$ such that

$$\hat{I} = f^{-1}[V^{-1}\{g^{-1}(C_{desired}) - F\}]\,. \tag{6}$$

where $\hat{I}$ is the compensated projector input image [3, 9].

The unknown parameters are computed as follows. Let *D* be the diagonal matrix with diagonal entries of *V*. Then we define a matrix $\hat{V} = VD^{-1}$ so that the diagonal entries of $\hat{V}$ are 1. Then,

$$C = g(\hat{V}D \cdot f(I) + F)\,. \tag{7}$$

$\hat{V}$ is estimated from the ratio of the changes in the three channels [3]. The camera nonlinear response function *g* is estimated from the images captured with different exposure time [5]. And the environmental lighting term *F* is measured by projecting a black image. With respect to an ideal screen, *D* becomes an identity matrix. Then, Equation (7) is written as

$$\hat{V}^{-1} \cdot g^{-1}(C) - F = f(I)\,. \tag{8}$$

Thus, the projector nonlinear response function *f* can be estimated from the projector input image *I* and the corresponding camera image *C*. The estimation is performed independently of the computation of the reflective property of the projection surface (*D*), which is efficient when the color or shape of projection surface is dynamically changing. Recall that *f* and *D* were considered as a nonlinear function together in [3], which is computationally inefficient.

In case of using N multiple projectors, Equation (7) is generalized as

$$C_i = g(\hat{V}_i D_i \cdot f_i(I_i) + F_i)\text{, } i\text{=1,2, …, } N. \tag{9}$$

Without loss of generality, it can be assumed that D_i and F_i are constant. Before compensation, all the projector input images are same as each other (I_1=I_2=…=I_N=I). Therefore, Equation (9) is written as

$$C_i = g(\hat{V}_i D \cdot f_i(I) + F)\text{, } i\text{=1,2, …, } N. \tag{10}$$

g, $\hat{V}_i$, f_i, and F can be estimated in advance. Therefore, once I and C_i are given, D (reflective property of the projection surface) can be estimated using Equation (10).

In a region of projection surface at a camera viewpoint, when the *k-th* projector is *diffuse projector,* only the *k-th* projector illuminates the region of projection surface [8]. Then, the projector input image of the *k-th* projector I_k is compensated in advance to obtain the desired camera image $C_{desired}$ such that

$$\hat{I}_k = f_k^{-1}[D^{-1}\hat{V}_k^{-1}\{g^{-1}(C_{desired}) - F\}]\,. \tag{11}$$

where $\hat{I}_k$ is the compensated projector input image.

5 Experimental Results and Discussion

Before the experiments, we measured a simplified isotropic BRDF to examine the reflection property of several typical materials (plywood, felt, paper, acrylic plastic, polyester film, stylene foam, cellophane). A projector (light source) projects a pure white image with a resolution of 1024 by 768 pixels at a fixed position (at an angle of 90 degree and at a distance of 1.2m) and a camera captures it at the different positions (from 0 to 180 degree at a distance of 1m) as shown in Figure 1. From the intensities of an identical region (51 by 51 pixels) at the camera images (see Figure 2), a curve of representing the reflection property of each material was obtained. Figure 3 shows the channel-wise reflection properties of the materials. Plywood or stylene foam has little specular reflection (close to Lambertian). On the contrary cellophane or acrylic plastic has excessive specular reflection. Especially, cellophane does not have sufficient diffuse reflection that the proposed method may not be useful for them. Polyester film is quite specular and also has sufficient diffuse reflection. Therefore, it is most suitable for verifying the performance of the proposed method.

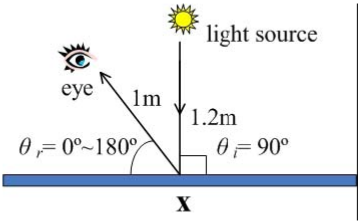


Fig. 1. Measurement of BRDF at a point. θ_i: incident angle, θ_r : reflective angle.

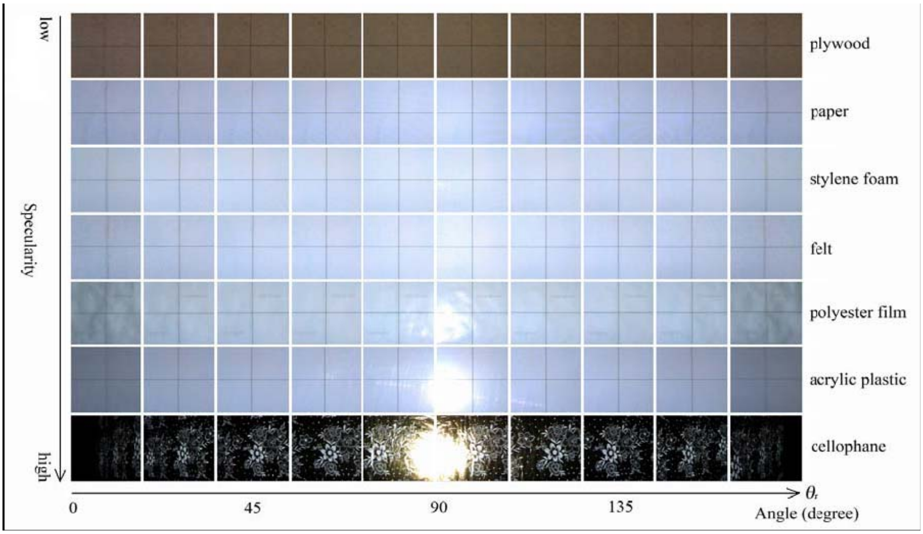


Fig. 2. Measurement of a simplified isotropic BRDF of several typical materials (plywood, felt, paper, acrylic plastic, polyester film, stylene foam, cellophane)

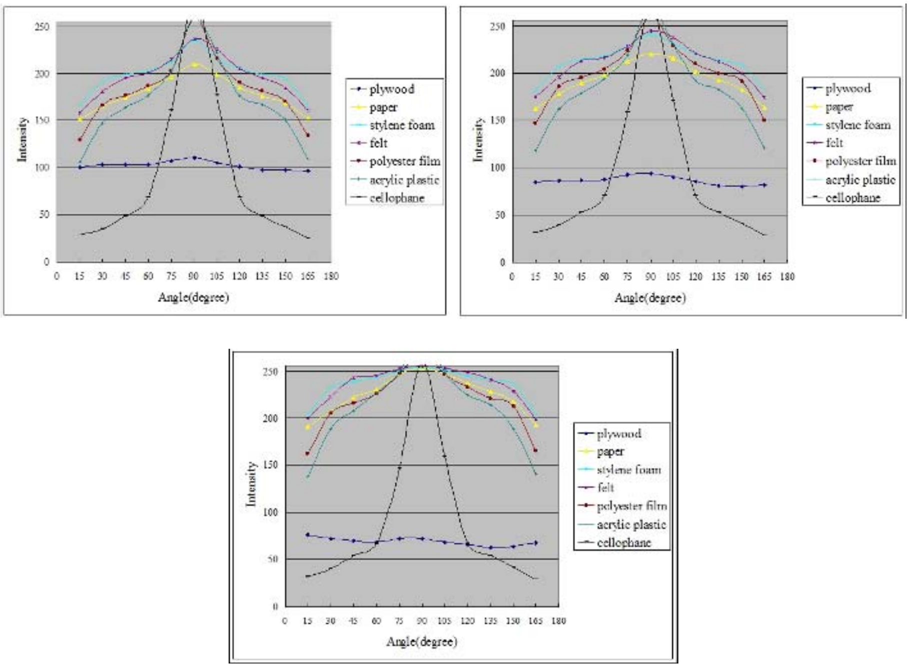


Fig. 3. Reflectance curve in RGB channels. Enumeration in order of their specularity: cellophane, acrylic plastic, polyester film, felt, stylene foam, paper, plywood. Left-top: red channel, left-bottom: blue channel, right: green channel.

Figure 4 shows our experimental setup. Two projectors (SONY VPL-CX6 [14]) and a single camera (PointGrey Dragonfly [10]) were used in our experiments. Two video cards (nVidia Geforce 6800LE, ATi Radeon 7000) with dual monitor support are connected with two monitors and two projectors separately. Two monitors are used to control the working program and the camera image. The images were at

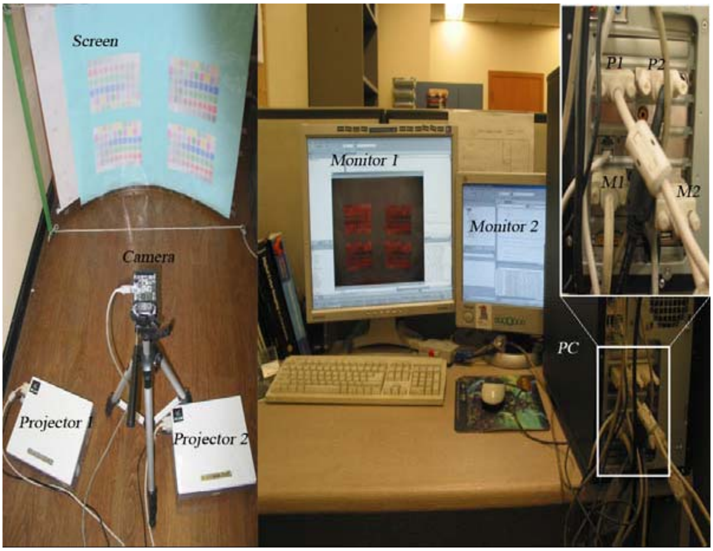


Fig. 4. Experimental setup. A camera is positioned between two projectors. Two video cards (nVidia Geforce 6800LE, ATi Radeon 7000) with dual monitor support are connected with two monitors and two projectors separately. Two monitors are used to control the working program and the camera image. The projection surface with multi-colored texture is covered with a polyester film which is quite specular.

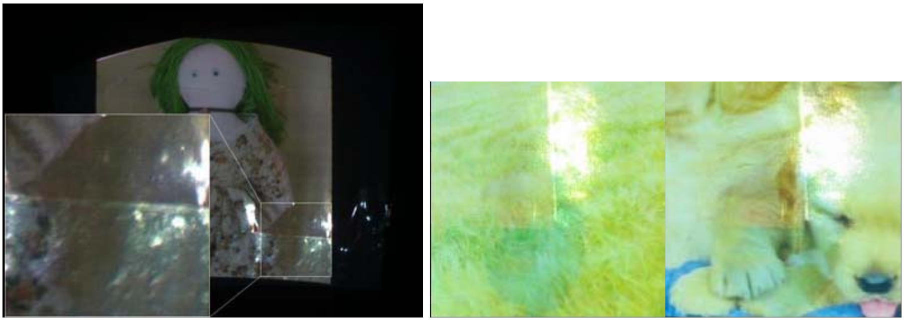


Fig. 5. Results of compensating a non-Lambertian surface using a single projector. In the resulting images, the specular region or its periphery was compensated wrongly. The specular region got stained after radiometric compensation. The right images are those of enlarging the specular region of some other resulting images.

a resolution of 1024 by 768 pixels. Screen texture used in our experiments is created by blurring and brightening a color chart because the radiometric compensation methods are not applicable to the screen with too much high texture or dark color.

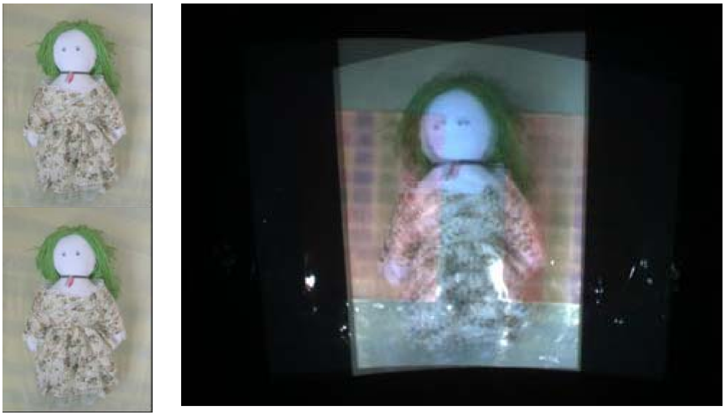
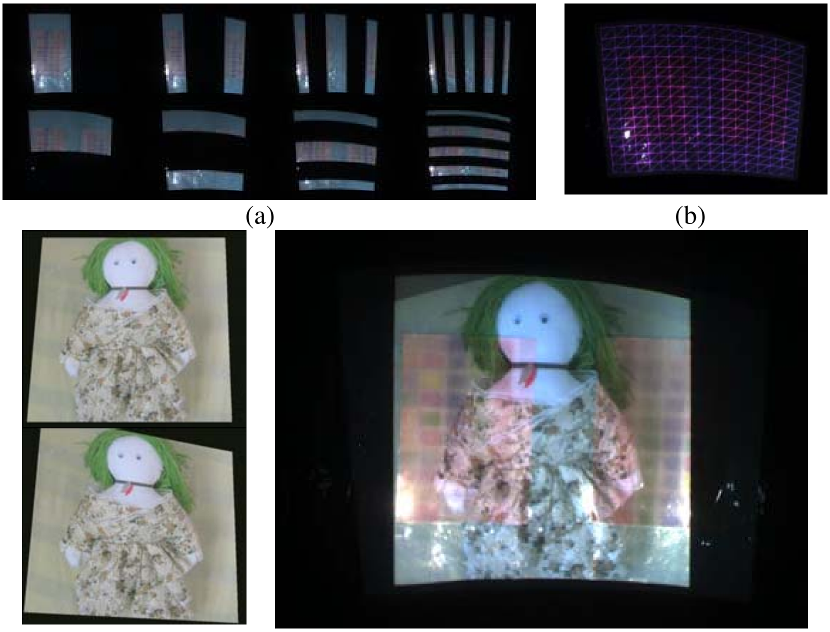


Fig. 6. Geometric and radiometric distortion. Left images: projector input images of two projectors, right image: projection by two projectors without any processing.



(a) (b)

(c) (d)

Fig. 7. Recovering the projection surface geometry using a binary code method and pre-warping the projector input images to be overlapped. (a) projecting the binary patterns onto the screen, (b) triangle meshes which were recovered using the binary-code method, (c) pre-warped projector input images to be overlapped, (d) overlapped projection.

The color chart includes various colors and thus is suitable for evaluating the performance of the radiometric compensation method. We covered the textured screen with a transparent polyester film to examine the problem due to specular reflection. For convenience and performance, some algorithms were implemented using OpenCV library [6] which includes most image processing and computer vision algorithms.

Figure 5 shows the results of applying the previous radiometric compensation method (using a single projector) [9] to a non-Lambertian curved surface. A part of the projection image was still distorted (not compensated) and got stained because the color information in the specular region is not readable in the camera view.

In case of using multiple projectors, without geometric registration, two projected images are distorted and not exactly overlapped because their coordinates are transformed by different projection matrix [8] as shown in Figure 6. Figure 7 shows the procedure of recovering the projection surface geometry using the binary-code method. The surface was represented as piece-wise planar with triangle meshes. After the regional homographies corresponding to the meshes were estimated, the projector input images were successfully pre-warped to be overlapped.

Figure 8 shows the results of applying the proposed radiometric compensation method (which is the same as the previous radiometric compensation method [9] except for utilizing multiple overlapping projectors) to a non-Lambertian curved surface. The color of the whole projection image was successfully compensated except for artifacts (which look like remaining specular reflection) due to the unevenness or imperfectness of the projection surface. Even if a pair of projector-camera system fails to compensate the color of a certain region of projection surface, the different projector-camera pair is used for successfully compensating the color of the region.

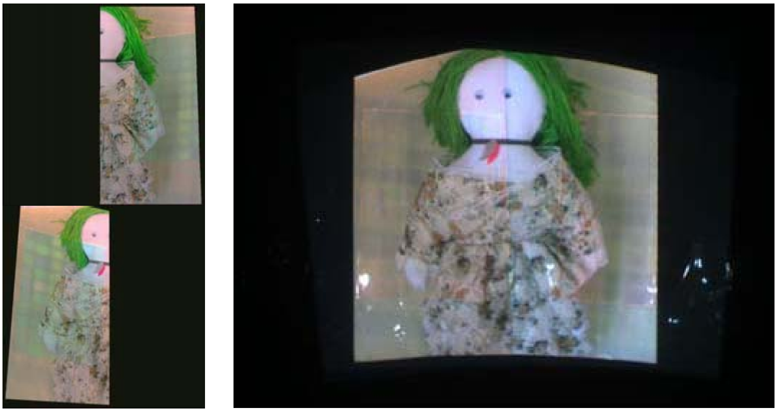


Fig. 8. An example of applying the proposed radiometric compensation method (using multiple overlapping projectors) to a non-Lambertian curved surface. Left images: projector input images which are partially compensated for compensating the whole image, right image: compensated projection image. The color of the whole projection image was successfully compensated except for artifacts due to the unevenness of the projection surface.

6 Conclusion

In this paper, it was demonstrated that multiple overlapping projectors could be successfully employed for radiometric compensated projection onto non-Lambertian surfaces. Reflectance of various non-Lambertian surfaces was examined experimentally. Except for highly specular materials (which do not usually have sufficient diffuse reflection) such as acrylic plastic and cellophane, the proposed method showed the performance far superior to existing radiometric compensation methods. Actually, the proposed method showed the best performance with respect to suitably specular materials with sufficient diffuse reflection such as polyester film.

Currently, we are trying to analyze the performance of the proposed method quantitatively.

In the future, it would be interesting to develop a radiometric compensation method of coping with a surface having more complicated reflection components (such as mutual reflection or subsurface scattering).

Acknowledgments. This study was supported by a grant(02-PJ3-PG6-EV04-0003) of Ministry of Health and Welfare, Republic of Korea.

References

1. Oliver Bimber, Franz Coriand, Alexander Kleppe, Erich Bruns, Stefanie Zollmann, and Tobias Langlotz. Superimposing pictorial artwork with projected imagery. IEEE Multimedia, pages 16-26, 2005.
2. Tat-Jen Cham, James M. Rehg, Rahul Sukthankar, and Gita Sukthankar. Shadow elimination and occluder light suppression for multiprojector displays. Proc. of CVPR'03, pages 513-520, 2003.
3. Michael D. Grossberg, Harish Peri, Shree K. Nayar, and Peter N. Belhumeur. Making one object look like another: controlling appearance using a projector-camera system. Proc. of CVPR'04, volume 1, pages 452-459, 2004.
4. Richard Hartley and Andrew Zisserman. Multiple View Geometry. Cambridge University Press, 2003.
5. Tomoo Mitsunaga and Shree K. Nayar. Radiometric self calibration. Proc. of CVPR'99, volume 1, pages 374-380, 1999.
6. OpenCV library. Available at http://sourceforge.net/projects/opencvlibrary/
7. Hanhoon Park, Moon-Hyun Lee, Sang-Jun Kim, and Jong-Il Park. Specular reflection elimination for projection-based augmented reality. Proc. of ISMAR'05, pages 194-195, 2005.
8. Hanhoon Park, Moon-Hyun Lee, Sang-Jun Kim, and Jong-Il Park. Specularity-free projection on nonplanar surface. Proc. of PCM'05, LNCS 3767, pages 606-616, 2005.
9. Hanhoon Park, Moon-Hyun Lee, Sang-Jun Kim, and Jong-Il Park. Surface-independent direct-projected augmented reality. Proc. of ACCV'06, LNCS 3852, pages 892-901, 2006.
10. PointGrey Dragonfly camera. Available at http://www.ptgrey.com/products/dragonfly/
11. Ramesh Raskar, Michael S. Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Seales, and Henry Fuchs. Multiprojector displays using camera-based registration. Proc. of Visualization'99, pages 161-168, 1999.

12. Ramesh Raskar, GregWelch, Kok-Lim Low, and Deepak Bandyopadhyay. Shader lamps: animating real objects with image-based illumination. Proc. of Eurographics Workshop on Rendering, pages 89-102, 2001.
13. Joaquim Salvi, Jordi Pages, and Joan Batlle. Pattern codification strategies in structured light systems. Pattern Recognition, 37(4):827-849, 2004.
14. SONY VPL-CX6 projector. Available at http://www.sonystyle.com
15. Rahul Sukthankar, Robert G. Stockton, and Matthew D. Mullin. Smarter presentations: exploiting homography in camera-projector systems. Proc. of ICCV'01, volume 1, pages 247-253, 2001.
16. Rajeev J. Surati. Scalable Self-Calibrating Display Technology for Seamless Large-Scale Displays. PhD thesis, MIT, 1999.

Motion Detection in Complex and Dynamic Backgrounds

Daeyong Park[1], Junbeom Kim[1], Jaemin Kim[1], Seongwon Cho[1], and Sun-Tae Chung[2]

[1] School of Electronics and Electrical Engineering, Hongik University,
72-1 Sangsu-dong, Mapo-gu, Seoul 121-791, Korea
tkc-tmhk@hanmail.net, jmkim@hongik.ac.kr, swcho@hongik.ac.kr
[2] School of Electronics Engineering, Soongsil University
cst@csu.ac.kr

Abstract. For the detection of moving objects, background subtraction methods are widely used. In case the background changes, we need to update the background in real-time for the reliable detection of foreground objects. An adaptive Gaussian mixture model (GMM) combined with probabilistic learning is one of the most popular methods for the real-time update of the complex and dynamic background. However, the probabilistic learning approach does not work well in high traffic regions. In this paper, we classify each pixel into four different types: still background, dynamic background, moving object, and still object, and update the background model based on the classification. For the classification, we analyze a sequence of frame differences at each pixel and its neighborhood. We experimentally show that the proposed method learn complex and dynamic backgrounds in high traffic regions more reliably, compared with traditional methods.

1 Introduction

Security on various regions (subways, airports, harbors, etc.) rapidly increases after 911-terror attack in USA. Traditional surveillance systems like a digital video recoding (DVR) system cannot detect an abnormal situation in real time. For the real-time and automatic detection of an abnormal situation, an intelligent procedure, which automatically detects events and is aware of situation, is required. An intelligent visual surveillance system is composed of many stages: motion detection, object classification, tracking, and behavior understanding [1]. Fig. 1 shows the various stages. In these stages, the motion detection stage is very important for accurate and reliable processing in subsequent stages. For motion detection, adaptive background subtraction is widely used, in which each pixel of the background is statistically modeled. Model accuracy is a heart of adaptive background subtraction especially when an environment has many changes. An adaptive Gaussian mixture model (GMM) is a reasonable approach for backgrounds with various changes because it well describes complex intensity distributions due to the changes and is mathematically easy to handle.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 545 – 552, 2006.

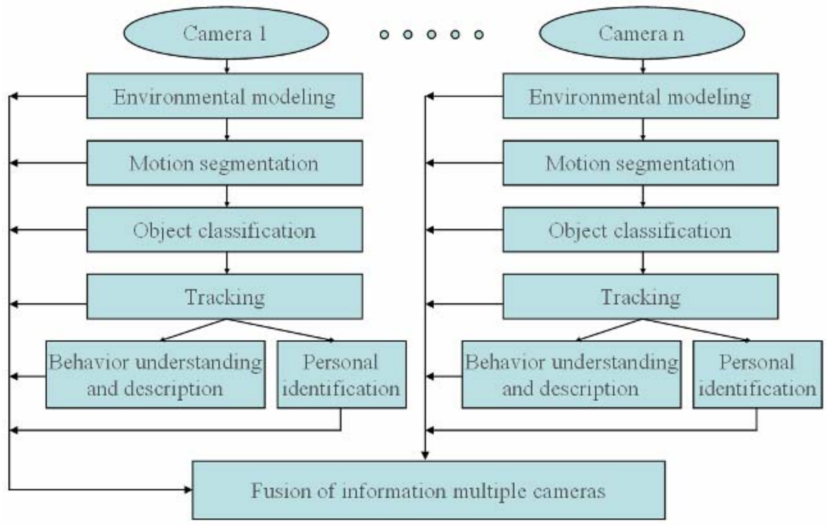


Fig. 1. General framework of visual surveillance

Most applications involving GMM use some variants of EM algorithm for the learning of the model parameters. In the learning, the rate of adaptation is important. Too large rate values make foreground be modeled as background. Too small rate values result in slow convergence whenever environment changes. In both two cases, detection is not reliable and it adversely affects next stages.

In the motion detection, Stauffer and Grimson [2] proposed a popular learning method for GMM. They used an online EM approximation based on a recursive filter to learn the mixture background model. The method can detect small objects using a slow global convergence rate. However, it causes slow convergence to the local changes of an environment. Most recently, Lee [3] proposed an effective learning method, which improves the convergence rate without compromising model stability. The method makes the learning rate slow down as the amount of learning increases: that is to say, a recently built-up Gaussian distribution, which has a small amount of learning, converges fast, and a long standing Gaussian distribution, which has a large amount of learning, converges slow. KaewTraKulPong and Bowden [4] proposed different methods for the learning rate adaptation. However, in high traffic regions, those approaches with only learning rate adaptation cannot detect moving objects reliably. The approaches assumed that in GMM, a Gaussian distribution for the background has larger values than those for moving objects. However, in a high traffic region, the Gaussian distributions for moving objects has a large variance value, and finally it is difficult to differentiate the Gaussian distribution for the background from those of foreground objects.

For reliable and robust detection of foreground objects, Haville, Goldon, and Woodfill [5] proposed a method using stereo vision. They use the fact that the depth of background is deeper than the depths of foreground moving objects. However, the estimation of depth using a stereo vision in outdoor is unreliable yet, is computationally complex, and needs more than one camera.

In this paper, we present a new learning method of a Gaussian mixture model for a background pixel. In the proposed method, we classify each pixel into four different types: a still background, a dynamic background, a moving object, and a still object, and update the GMM appropriately based on the classification result. For the classification, we use a temporal motion history like a sequence of frame differences in addition to a likelihood test.

In Section 2, we describe the detection of foreground objects. In the following section 3, we compare the proposed method with two traditional methods using three difference test videos. We draw conclusions in Section 4.

2 Detection of Foreground Objects

In the learning of the background, we first detect pixels with motion, and then classify the pixels into a dynamic background or a moving object. When a pixel is in a still region, we classify it into a still background or a still foreground object. We model the background using a Gaussian mixture model. Based on the model, we compute the likelihood of the pixel belonging to the background and classify each pixel. Once we classify the pixel into a background, we update the model parameters for the background. Fig. 2 shows the overall procedure.

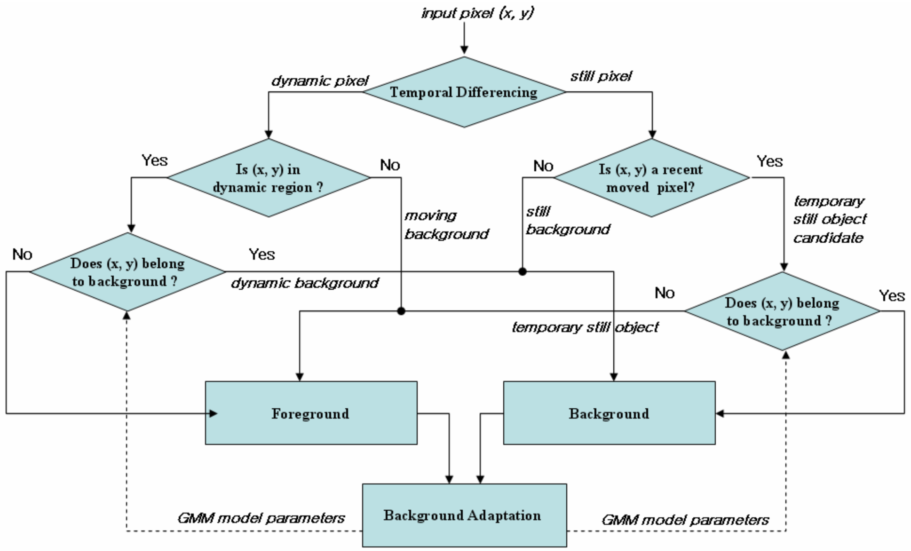


Fig. 2. The overall procedure of the proposed learning method

For the detection of a moving object in the proposed procedure, we extend the method proposed by Collins [6]. We briefly summarize the method in the subsection 2.1.

2.1 Detection of Moving Object

Using temporal frame difference, we can detect moving objects. When an object moves slowly, a single temporal frame difference show only the partial region of the moving object. For the reliable detection of the silhouette of a moving object, we need many frame differences. Collins defines a motion trigger, T and a stability measure, S, which are based on the history of frame differences. Two parameters are defined as

$$T = \max\{ I(t) - I(t-j)\,|, \forall j \in [1, N]\} \tag{1}$$

$$S = \frac{M \sum_{j=1}^{M} I(t-j)^2 - \left(\sum_{j=1}^{M} I(t-j) \right)^2}{M(M-1)} \tag{2}$$

When two parameters are larger than given threshold values, respectively, a pixel (x,y) is detected as a moving object pixel, and the status of the pixel, M(x,y), is set to 1. Otherwise, it is set to zero. The threshold should be higher than the values due to illumination changes.

2.2 Detection of Temporary Still Object

When an object stops moving for a short period or most region of the object has similar intensity, two parameters, T and S, have small values and we detect it as a background. In the proposed method, when a pixel status M(x, y) changes from one to zero at a frame, we choose the pixel as a candidate for a temporary still object for some period. During this period, if the pixel value is within some range, we classify it as the background and update the model parameters for the background. We determine the range using the recent means and variances of Gaussian mixture models. Otherwise, we classify it a temporally still object or noise. We update the model parameter using the method proposed in [3].

2.3 Detection of Dynamic Background

In a waving jet of water or a moving escalator in the background, the intensity change is larger than the changes due to illumination change or noises. As a result, two parameters, T and S, have large values, and we detect it as a moving object. Under the assumption that moving objects pass by a region in a short time period, if M(x, y) is set to 1 for some period of time, we need to determine whether the pixel (x,y) belongs to a dynamic background or not. Since we can describe most dynamic background well with a Gaussian mixture model, we can classify the pixel type with the likelihood test. If the pixel belongs to a dynamic background, we update the model parameters. We update the model parameter using the method proposed in [3]. For reliable operation, we need to learn the model parameters for a long period of time when there is no traffic.

2.4 Still Background

When a pixel is in a still region, we classify it into a still background or a still foreground object. The pixel values in the still background region changes slightly by illumination changes or camera noises. Based on the off-line trained GMM parameters, means and variances, we compute the likelihood of the pixel belonging to the background and classify each pixel. Once we classify the pixel into the still background, we update the model parameters for the still background. We update the model parameter using the method proposed in [3].

3 Experimental Results

We tested the proposed method using two different videos: high traffic region and high traffic region with a dynamic background (a moving escalator). The frame rate is 15 frames per second in all videos.

"High traffic region" is acquired at a passageway of a department store. It consists of 900 image frames and about 20 people passed by in the video. The first column in Fig. 3 shows the last frame.

"High traffic region with a dynamic background" is acquired in a subway. It contains a moving escalator modeled as a dynamic background. It consists of 660 frames and 18 people passed by in the video. The second column in Fig. 3 shows the 660th frame.

We compared the proposed method with the works of Stauffer and Lee using the three videos. In the experiments, we used a Gaussian mixture model. We set the initial converge rate 0.05 and used the Bayesian frame classifier for motion detection described in [3]. The same sigmoid function was used (a=96, b=3). Table 1 shows the comparison of the learned model parameter values at the pixels marked with red crosses in Fig. 3.

In the experiments, the method of Stauffer [2] is relatively stable in high traffic regions because of its slow convergence rate. In all test videos, the model variance does not converge to an appropriate value and remains near the initial value 40, which is shown at Table 1. As a result, it cannot model the background properly especially when the foreground objects have similar color. The method of Lee [3] learns the background model properly with a fast convergence rate in general environments. However, in high traffic region, the model parameters wander widely. As the result, the model parameters have large variances and cannot model the background properly.

Generally, the proposed method learned the background model with a fast convergence rate in high traffic regions as well as in general environments. In two videos, the proposed method prevents the foreground object corrupting the background model. As a result, the learned variance has small values, compared with other two methods. In these cases, the proposed method can detect the foreground objects more accurately than two other methods.

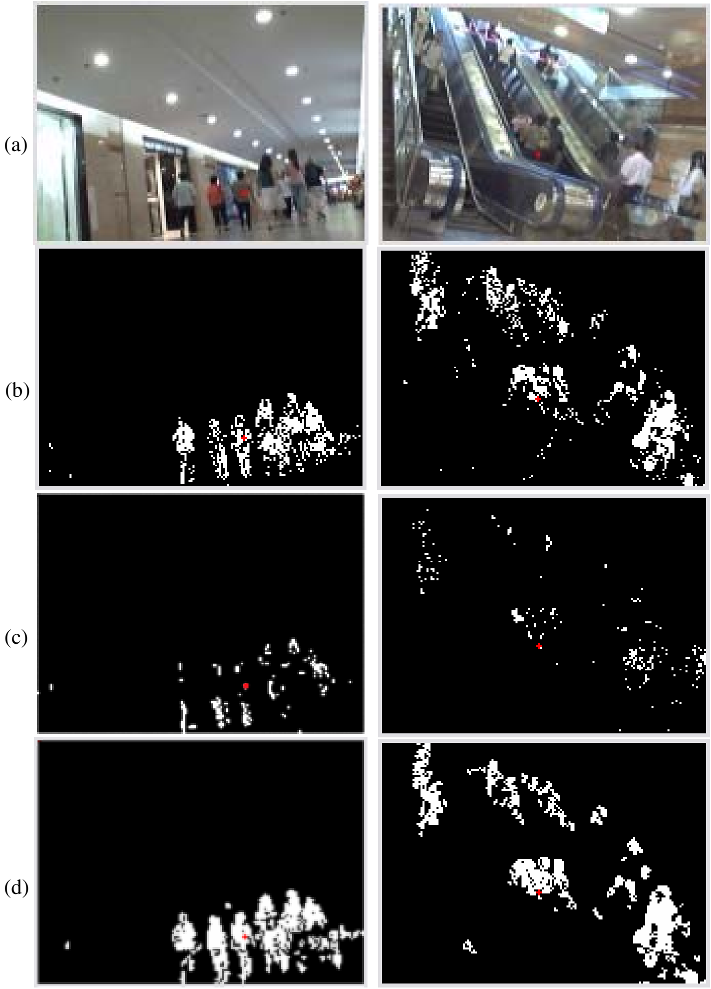


Fig. 3. Comparison of motion detection results in two different videos with high traffics: The row (a) shows two different videos. Their frame numbers are 900 and 664, respectively. The row (b) and (c) show the detected foreground using the methods described in [2] and [3], respectively. The row (d) shows the results of the proposed method.

Table 1. Comparison of the learned model parameter values at the pixels marked with red crosses in Fig. 3

		The method of [2]			The method of [3]			The proposed method		
Gaussian No.		Y1	Y2	Y3	Y1	Y2	Y3	Y1	Y2	Y3
Video A	Mean	90.21	58.00	23.99	86.51	-	-	92.09	22.06	72
	Variance	37.93	39.98	39.40	200	-	-	15.68	11.68	40
	C	627.00	8.00	109.00	899.00	-	-	364	52	1
	Weight	0.85	0.04	0.11	1.00	-	-	0.936	0.058	0.004
Video B	Mean	52.97	139.00	83.08	87.92	-	-	67.94	88.67	46.13
	Variance	40.07	40.02	40.21	389.16	-	-	31.09	17.68	17.23
	C	117.00	14.00	410.00	618.00	-	-	175.04	50.45	68.50
	Weight	0.19	0.05	0.76	1.00	-	-	0.66	0.22	0.13

4 Conclusion

Adaptive Gaussian mixtures models are widely used for modeling complex backgrounds for motion detection. In this paper, we pointed out a limitation of traditional probabilistic learning methods in high traffic regions, and presented an effective learning algorithm. The presented method improved the model learning with the classification of each pixel into four different types. The classification was based on the temporal history of the pixel intensity variation. With experiments, we showed that the proposed method significantly enhanced foreground object detection in high traffic regions, compared with two traditional methods.

Acknowledgement

This work was supported by 2006 Hongik University Research Fund, the Soongsil University Research Fund, BK21, and Seoul Metropolitan government.

References

1. Hu, W., Tan, T., Wang, L., Maybank, S.: A Survey on Visual Surveillance of Object Motion and Behaviors. In: IEEE Trans. on Systems, Man, and Cybernetics. Part C: applications and reviews, Vol. 34, No 3. (2004) 334-351
2. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition. (1999) 246-252

3. Lee, D.: Effective Gaussian Mixture Learning for Video Background Subtraction. In: IEEE Trans. on Pattern Analysis and Machine Intelligence Vol. 27. (2005) 827-832.
4. KaewTraKulPong, P., Bowden, P.: An improved adaptive background mixture model for real-time tracking with shadow detection. In: Proceedings of Second European Workshop on Advanced Surveillance System. (2001) 149-158
5. Harville, M., Gordon, G., Woodfill, J.: Foreground Segmentation Using Adaptive Mixture Models in Color and Depth. In: Proc. of International Conference on Computer Vision Workshop Detection and Recognition of Events in Video. (2001) 3-11
6. Collins, R.T., Lipton, A.J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt, P., Wixson, L.: A System for Video Surveillance and Monitoring. In: CMU-RI-TR-00-12, Carnegie Mellon University (2000)

Effective Face Detection Using a Small Quantity of Training Data

Byung-Du Kang, Jong-Ho Kim, Chi-Young Seong, and Sang-Kyun Kim

Department of Computer Science, Inje University, Gimhae, 621-749, Korea
{deweyman, luckykjh, cy1224, aiskkim}@gmail.com

Abstract. We present an effective and real-time face detection method based on Principal Component Analysis (PCA) and Support Vector Machines (SVMs). We extract simple Haar-like features from training images that consist of face and non-face images, reinterpret the features with PCA, and select useful ones from the large number of extracted features. With the selected features, we construct a face detector using an SVM appropriate for binary classification. The face detector is not affected by the size of a training dataset in a significant way, so that it works well with a small quantity of training data. It also shows a sufficiently fast detection speed for it to be practical for real-time face detection.

1 Introduction

Face detection has been used in various fields such as security systems with user identification and user context-aware systems with both specific user recognition and tracking. In these fields, face detection is the most fundamental and significant step, so that performance of the entire system is dependent on whether the face detection algorithm succeeds or not. Therefore, various theories have been actively studied to detect faces effectively and in real time. Face detection is a problem of binary classification that divides images into two regions: a face region and a non-face region. While many successful research results for face detection have been presented in the literature, there has been little progress for its real application because of its complexity. Faces have inherent differences such as color, size, and shape. They have various state changes that result from facial expressions, hairstyles, facial make-up, and accessories like glasses. They are also affected by external environments such as lighting effects and complex backgrounds. Several attempts have been made to deal with these obstacles to face detection.

These methods can be divided into facial feature-based approaches and facial shape-based approaches. Graf[1] suggested a method to locate faces with a threshold value after intensity enhancement of the gray image. This method has difficulty handling lighting changes. Cail[2] suggested a method that uses skin color information defined in the color space of CIE Lab. Although it takes advantage of color information instead of simple gray information, it is affected by changes of lighting. Craw[3] used templates that describe facial features. This method improves frame rates, but it cannot process the changes of facial size, rotation, and pose. These methods are easy

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 553 – 562, 2006.

to implement and detect faces rapidly with simple operations. However, they do not consider many variations of face images. To solve these problems, statistical approaches based on neural networks, statistical distributions, and AdaBoost were suggested. Among them, neural network-based approaches use the adapted structure of a neural network such as hierarchical neural network[4], auto associative neural network[5], probabilistic decision-based neural network(PDBNN)[6], and convolutional networks[7]. A typical method of this approach is that of Rowley[8], which introduces the concept of arbitration. This method is good for training and modeling of complex face patterns, but it is slow as it has many operations. Sung and Poggio[9] suggested a method to get faces and backgrounds based on the distribution of Gaussian clusters in a high dimensional space. This method ensures high detection rates only if the training data for backgrounds are appropriate. However, it is difficult to collect typical samples for various backgrounds. Finally, these methods have problems of both sampling and frame rates. To ensure high frame rates and high efficiency of detection simultaneously, Viola and Jones[10] suggested a face detection method using the AdaBoost algorithm, which reduces the number of complex operations. Operating on 384×288 pixel images, their detector processes 14 frames per second with high detection rates. However, to get feasible performance, they need a big enough dataset of face and non-face images for training the detector. They also have to retrain and reconstruct the entire system for additional data even though they use large training datasets. These problems are common to AdaBoost-based approaches.

In this paper, we propose a method that has high detection rates and acceptable frame rates when applied to a real-time system with only a small quantity of training data. In real-time face detection, complex features that require many operations cannot be used because of the impact on the frame rates. We use simple Haar-like features used by Viola and Jones[10] to achieve high frame rates. However, it is difficult to obtain useful features to characterize faces and non-faces from the large number of simple features. So, Viola et al. use a cascade structure that eliminates the background by stages with specific groups of simple features selected for each step. To deal with this problem, we extract the useful features with a Principal Component Analysis (PCA), which analyzes correlations between simple features and transforms feature space into principal component space. The selected features from the principal component space are used as feature vectors of a Support Vector Machine (SVM), which estimates a class of populations with them. The SVM classifier also provides high performance on the binary classification problem. Therefore, our detector works well with only a small quantity of training data.

In this experiment, our detector showed high detection rates and acceptable frame rates. Even though frame rates are lower than those of Viola and Jones[10], it can process eight frames per second for 320×240 pixel images. This is an acceptable processing time for a real-time system.

2 Face Detection Using PCA and SVM

Fig. 1 shows the main structure of our face detection system. The data collector accumulates the value of Haar-like features. In the second step, the feature space is transformed into the space of the principal components. The selected features from the principal component space are used as feature vectors for the SVM. In the next step, the SVM classifier is trained with the training patterns. In the last step, the SVM classifier classifies regions into faces and non-faces.

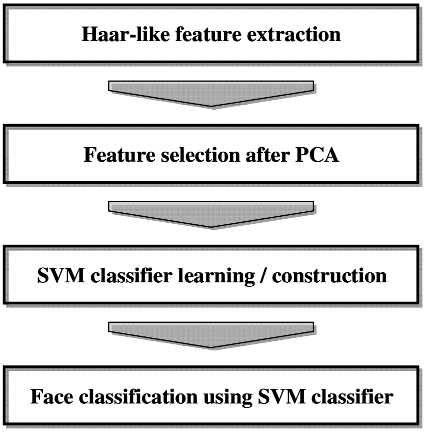

Haar-like feature extraction
Feature selection after PCA
SVM classifier learning / construction
Face classification using SVM classifier

Fig. 1. Four steps of the face detection

2.1 Feature Extraction

The face detector is based on the simple rectangle features presented by Viola and Jones[10], who measured the differences between region averages at various scales, orientations, and aspect ratios. The rectangle features can be evaluated extremely rapidly at any scale (see Fig. 2). However, these features require very large training datasets. Therefore, after analyzing principal components, we select useful features from each of five rectangle features. These selected features are used as feature vectors for the SVM. Experiments demonstrate that they provide useful information and improve performance of accurate classification with only small amounts of training data.

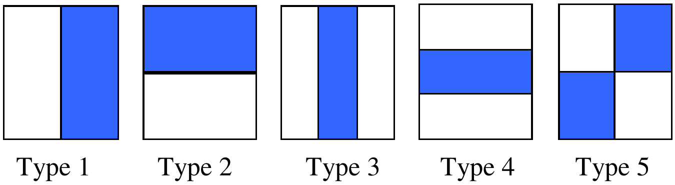


Fig. 2. Used Haar-like features

Fig. 3 shows 100 face images and 100 non-face images used for PCA. The non-face images are selected carefully to describe various backgrounds and human bodies.

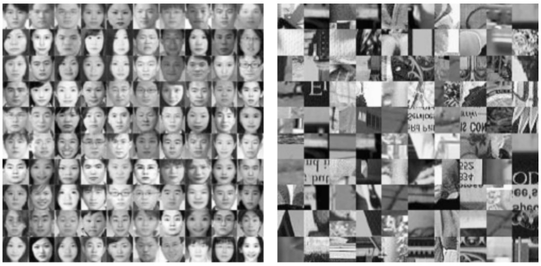
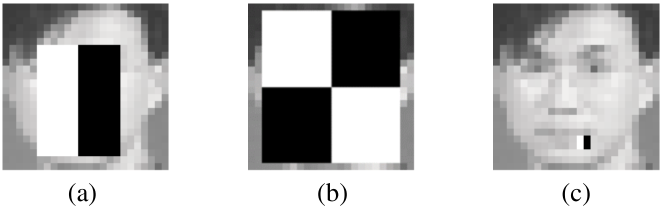


Fig. 3. Face and non-face images for PCA

In the selection of useful features, we regard the following features as useless and eliminate them before performing PCA.

1. A feature value near 0, which results from the symmetry structure of the face (Fig. 4. a).
2. The size of a feature being more than 450 pixels (Fig. 4. b).
3. The size of a feature being less than 18 pixels (Fig. 4. c).

Figure 4 shows these useless features.



(a) (b) (c)

Fig. 4. Useless features

The Haar-like features on the image are transformed to the space of principal components using PCA, which is a method of multivariate statistical analysis.

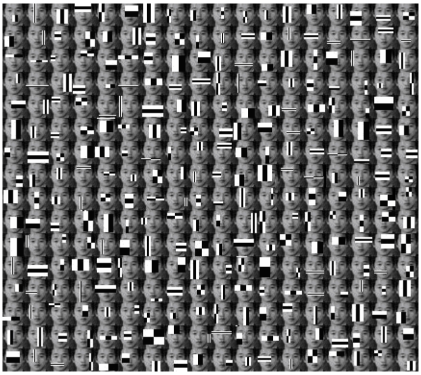


Fig. 5. The 288 useful features selected

In this paper, we used 12 principal components that explain features with more than the cumulative explanation rate of 90%. From the 12 principal components, we selected 288 useful features from all possible 162,336 Haar-like simple features. Fig. 5 shows 288 useful features selected using PCA. Consequently, a training image is converted to 288 values corresponding to the chosen useful features, and our SVM classifier uses this input vector of 288 dimensions for training.

2.2 SVM Classifiers

SVM classifiers find a separating hyperplane that maximizes the margin between two classes, where the margin is defined as the distance of the closest point, in each class, to the separating hyperplane (see Fig. 6). This is equivalent to performing structural risk minimization to achieve good generalization[13].

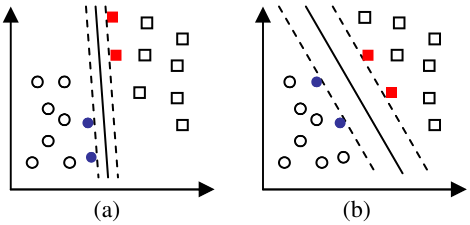


Fig. 6. (a) A separating hyperplane with small margin. (b) A separating hyperplane with a large margin. The filled circles and rectangles are termed "support vectors."

Given a dataset $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$ of l examples $\mathbf{x}_i$ with labels $y_i \in \{-1,+1\}$, finding the optimal hyperplane implies solving a constrained optimization problem using quadratic programming, where the optimization criterion is the width of the margin between the classes. The separating hyperplane can be represented as a linear combination of the training examples, and classifying a new test pattern $\mathbf{x}$ is done using the following expression:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b_i \tag{7}$$

where $k(\mathbf{x}, \mathbf{x}_i)$ is a kernel function and the sign of $f(x)$ determines the class membership of $\mathbf{x}$. Constructing the optimal hyperplane is equivalent to finding the a nonzero α_i. Any data point $\mathbf{x}_i$ corresponding to nonzero α_i is termed a support vector. Support vectors are the training patterns closest to the separating hyperplane, and the kernel function extends the SVM to handle nonlinear separating hyperplanes. Popular kernel functions include, $k(\mathbf{x}, \mathbf{x}_j) = \exp(-\|\mathbf{x} - \mathbf{x}_j\|^2 / 2\sigma^2)$ which leads to a Gaussian RBF, and $k(\mathbf{x}, \mathbf{x}_j) = (\mathbf{x}^T \mathbf{x}_j + 1)^d$ which is a polynomial of degree d.

Fig. 7 gives an overview of detection progress that includes simple feature extraction, feature analysis, and classifier construction. Firstly, from the Haar-like simple features, useful features are selected using PCA. Training images, as shown in Fig. 8, are converted to input vectors of 288 dimensions with selected features. The SVM classifier uses these input vectors for training.

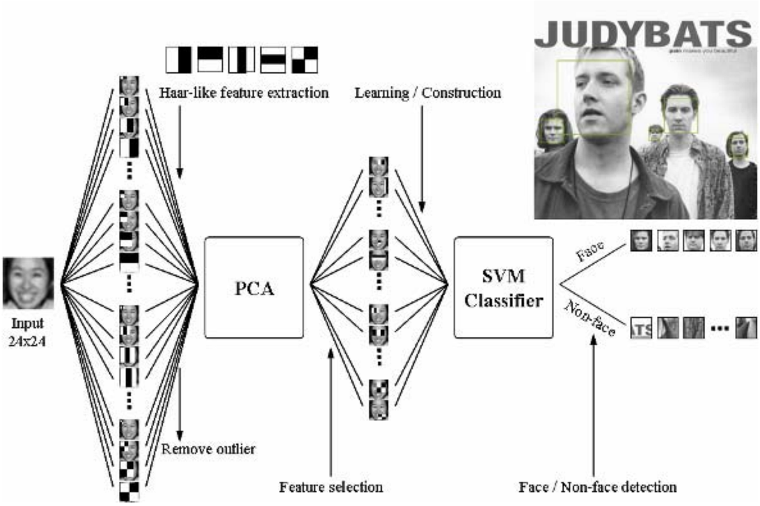
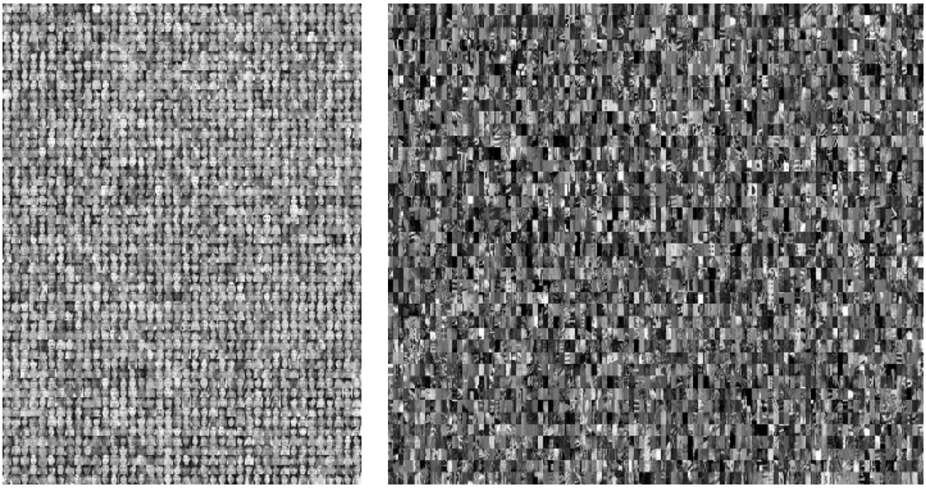


Fig. 7. The basic structure for the face detection

We collected training data from various sources such as the Internet, a digital camera, a PC-cam, and a well-known face database. To reduce the misclassification rate due to various backgrounds, we constructed a training set for non-face images that was twice as large as the training set for face images.



(a) Face images (b) Non-face images

Fig. 8. Training data for the SVM

To detect face of various sizes, there are two methods. One is to vary the size of the training image itself. The other is to normalize input images of various sizes to a fixed size of training image. As the first method requires many operations to change the size of training images, we instead normalized input images to a fixed size of 24×24 pixels for real-time detection.

3 Experimental Results

Our face detector was implemented using Visual C++ on a 2.0 GHz Pentium IV PC with the Microsoft Windows operating system. We experimented on detection rates and frame rates according to a quantity of training data. To compare the performance of our detector with that of related works, we also tested our detector with the CMU test set. Fig. 9 shows examples of face detection.

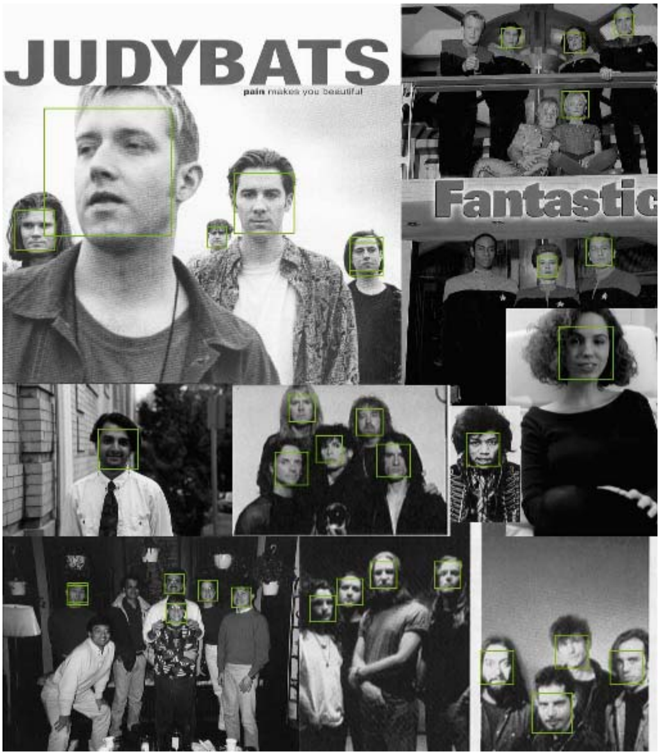


Fig. 9. Examples of face detection

We also did experiments to see the effect on face detection performance of changing the number of training images that influence the quantity of training data and the number of features that is used in training. The training images include 1000 face images and 1000 non-face images.

Fig. 10 shows face detection results using 100, 200, 500, and 1000 training images. These show our detector is flexible with respect to the quantity of training data and works well with only a small quantity of training data.

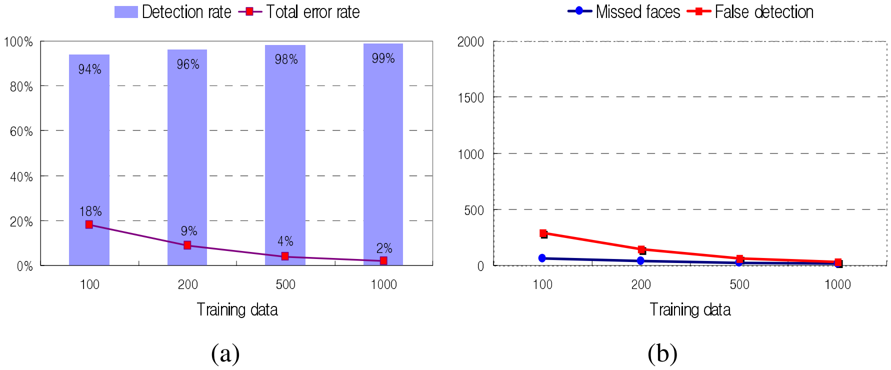


Fig. 10. Face detection results according to the amount of training data

Fig. 11 shows face detection results from when we used Haar-like features of 100, 200, and 500 features. With the relatively small number of 100 selected features, a good result of 94% was obtained.

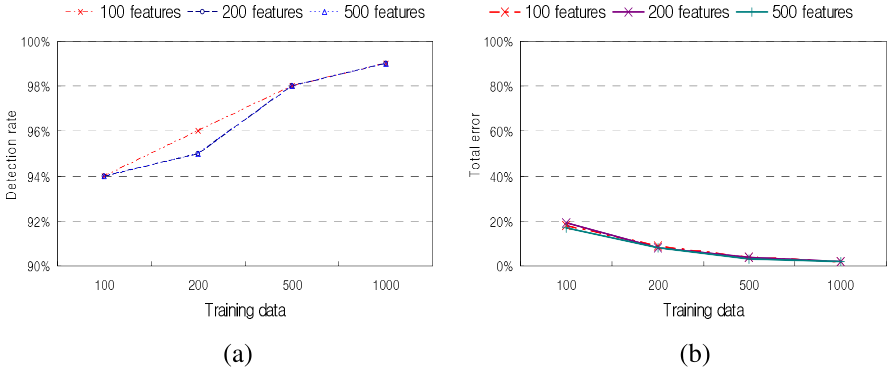


Fig. 11. Face detection results according to the number of features

Consequently, the proposed method showed good face detection with only a small quantity of training data. It can also process eight frames per second for 320×240 pixel images. Although these frame rates are lower than those of Viola and Jones[10], it is an acceptable processing time for real-time systems.

3.1 Comparison with Related Works

We compared our detector with that of related works through a test with the CMU dataset, which consists of 130 gray-scale images and contains 507 faces (see Table 1).

Table 1. Detection rates with the CMU test set

Face detectors	Missed faces/total faces	Detection rate	False alarms
Feraud-et[5]	73/507	85.6%	8
Garcia-Delakis[7]	49/507	90.3%	8
Rowley-Baluja-Kanade[8]	83/507	83.6%	10
Viola-Jones[10]	96/507	81.1%	10
Using PCA and SVM	50/507	90.1%	8

We can not compare the processing time of the other researchers' algorithms precisely because of the difficulty of testing each algorithm in the same system environment. Table 2 shows results of each study presented in their papers and our result.

Our detector got a detection rate of 90.1% and a detection speed of 0.12 second per frame with the CMU dataset. This is better than Viola-Jones[10] from the viewpoint of detection rate and Garcia-Delakis[7] from viewpoint of the detection speed. Furthermore, it showed eight false alarms, which is lower than that of Viola-Jones[10], and had a high detection rate and acceptable detection speed.

Table 2. Processing time of related works

Face detectors	Processing time(s)	Test environments
Feraud-et[5]	1.27 s	Average processing time with the CMU test set
Garcia-Delakis[7]	0.25 s	1.6 GHz Pentium IV processor
Rowley-Baluja-Kanade[8]	7.2 s	200 MHz R4400 SGI Indigo 2
Viola-Jones[10]	0.067 s	700 MHz Pentium III processor
Using PCA and SVM	0.12 s	2.0 GHz Pentium IV processor

4 Conclusion

In this paper, we proposed a face detection method that is not affected by the size of a training dataset and gets a sufficiently good detection speed for use in real-time face detection. We used simple Haar-like features for detection speed. They are selected with PCA to reflect characteristics of the population with only a small amount of training data. Using these feature vectors, we constructed a detector based on a SVM that provides high performance with binary classification problems.

In this experiment, our detector showed a good face detection rate of 90.1% and a frame rate of eight frames per second.

Although the proposed method detects faces efficiently, its performance will be enhanced if a tracking method was also attached. Furthermore, if 3D technology with several cameras is applied, it should deal well with the rotation of faces and could be used in many applications such as in a smart home.

Acknowledgement

This work was supported by the 2005 Inje University research grant.

References

1. Graf, H.P., Chen, T., Petajan, E. and Cosatto, E.: Locating faces and facial parts. Proceedings of First International Workshop Automatic Face and Gesture Recognition (1995) 41-46
2. Cai, J., Goshtasby, A. and Yu, C.: Detecting human faces in color images. Image and Vision Computing, Vol. 18 (1999) 63–75
3. Craw, I., Ellis, H. and Lishman, J.: Automatic extraction of face features. Pattern Recognition Letters, Vol. 5 (1987) 183-187
4. Agui, T., Kokubo, Y., Nagashashi, H. and Nagao, T.: Extraction of face recognition from monochromatic photographs using neural networks. Proceedings of Second International Conference on Automation, Robotics, and Computer Vision, Vol. 1 (1992) 1-5
5. Feraund, R., Bernier, O.J., Viallet, J. E. and Collobert, M.: A fast and accurate face detector based on neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23 (2001) 42-53
6. Lin, S. H., Kung, S. Y. and Lin, L. J.: Face recognition/detection by probabilistic decision-based neural network. IEEE Transaction on Neural Networks, Vol. 8 (1997) 114-132
7. Garcia, C. and Delakis, M.: Convolutional face finder: A neural architecture for fast and robust face detection. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 26 (2004) 1408-1423
8. Rowley, H., Baluja, S. and Kanade, T.: Neural network-based face detection. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 20 (1998) 23-38
9. Sung, K. K. and Poggio, T.: Example-based learning for view-based human face detection. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 20 (1998) 39-51
10. Viola, P. and Jones, M.: Robust real-time face detection. International Journal of Computer Vision, Vol. 57 (2004) 137-154
11. Johnson, R.A. and Wichern, D. W.: Applied Multivariate Statistical Analysis. Prentice Hall (2002) 356-395
12. 12 Turk, M. and Pentlan, A.: Face Recognition Using Eigenfaces, IEEE Conference on Computer Vision and Pattern Recognition (1991) 586-591
13. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)

A New Passage Ranking Algorithm for Video Question Answering

Yu-Chieh Wu[1], Yue-Shi Lee[3], Jie-Chi Yang[2], and Show-Jane Yen[3]

[1] Department of Computer Science and Information Engineering, National Central University
[2] Graduate Institute of Network Learning Technology, National Central University,
No.300, Jhong-Da Rd., Jhongli City, Taoyuan County 32001, Taiwan, R.O.C.
bcbb@db.csie.ncu.edu.tw, yang@cl.ncu.edu.tw
[3] Department of Computer Science and Information Engineering, Ming Chuan University,
No.5, De-Ming Rd, Gweishan District, Taoyuan 333, Taiwan, R.O.C.
{leeys, sjyen}@mcu.edu.tw

Abstract. Developing a question answering (Q/A) system involves in integrating abundant linguistic resources such as syntactic parsers, named entity recognizers which are not only impose time cost but also unavailable in other languages. Ranking-based approaches take the advantage of both efficiency and multilingual portability but most of them bias to high frequent words. In this paper, we propose a new passage ranking algorithm for extending textQ/A toward videoQ/A based on searching lexical information in videos. This method takes both *N*-gram match and word density into account and finds the optimal match sequence using dynamic programming techniques. Besides, it is very efficient to handle real time tasks for online video question answering. We evaluated our method with 150 actual user's questions on the 45GB video collections. Nevertheless, four well-known but multilingual portable ranking approaches were adopted to compare. Experimental results show that our method outperforms the second best approach with relatively 25.64% MRR score.

1 Introduction

With the rapid expansion of video media sources such as news, TV shows, and movies, there is an increasing demand for automatic retrieval and browsing of video data. Generally speaking, the most effective way of managing video data is to support a video document retrieval based on the input keyword queries. In a modern video retrieval scenario, users would query the system with short or natural language questions with his familiar language, e.g., "Where is the beginning of the Chinese culture?" or "In China, what is the most exquisite pottery city?" They expect the system to return short answers rather than the whole videos which may be in different language.

To support above goals, this implies several research fields such as, video content extraction, information processing, and question answering. Extracting contents in videos is a very difficult but complex task and becoming an important issue. Textual, visual, audio information is most frequently adopted features for this purpose. Among

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 563 – 572, 2006.

them, text in videos, especially for the closed captions is the most powerful of high-level semantics since it is closely related to current content and state-of-the-art optical character recognition (OCR) techniques are far more robust than existing speech or visual object analysis approaches [11]. Thus, almost all video content retrieval researches start with video text recognition [3] [8] [18] [21]. The well-known Informedia and TREC-VID projects are the typical examples.

Over the past few years, several related research studies had been proposed and addressed the use of either videoOCR or speech recognition (SR) techniques to support video question answering. Lin et al. [9] showed an earlier work on combining a simple videoOCR and lexical term weighting methods. They focused on extracting the "white" text components in image and hand-created keyword list to increase the lexicon. Besides, they also presented three strategies for improving OCR errors. In 2003, Yang et al. [19] proposed a complex videoQ/A system by integrating abundant resources such as, WordNet, Internet, shallow parsers, named entity taggers, and human-made rules. They employed the term weighting approach as [12] to rank the video shots that contain high-frequent related words. Usually when porting this system into another language or domain, these components should be re-trained by means of amount of annotated corpus which is often a huge work, for example building the treebank bracketing corpora. Cao et al. [2] proposed a domain-dependent video question answering system to enable learners engaging in the learning process. Unlike previous literatures, Cao used the pattern-matching approaches to pinpoint answers where pattern set was constructed by a domain expert. In the same year, Wu et al. [18] presented a cross-language videoQ/A framework based on the videoOCR techniques as mentioned in [6] [9]. They convert auto-translated Chinese OCR transcripts into English and performed the English Q/A method based on combining named entity taggers, mapping rules, and WordNet. Alternatively, Zhang and Nunamaker [21] treated the video clips as answers and applied the TFIDF (term frequency*inverse document frequency) term weighting schema to retrieve the manual-segmented clips. In that approach, they also hand-created the ontology and combined rich resources, such as parsers, and WordNet.

In this paper, we present a new passage ranking algorithm for extending textQ/A toward videoQ/A based on searching text information in videos. Users interact with our videoQ/A system through natural language questions with content of the expected videos. Our system retrieves relevant video fragments and supports both visual and text outputs in response to the user's questions. We consider that a passage is able to answer the question and suitable for videos since itself forms a very natural unit of response for question answering. Lin et al. [10] showed that users prefer passages over exact answer-phrase because paragraph-sized chunks provide context. In contrast to previous studies [2] [9] [18] [19], the proposed method is relatively simple yet and effective.

2 System Architecture and Video Processing

The overall architecture of the proposed video Q/A system is shown in Fig. 1. At first, the video processing module recognizes the caption words as transcript through the text localization, extraction&tracking, and OCR. Second, the Q/A processing module ranks the segmented passages in response to the input question. In this section, we describe the overall videoOCR processing. Section 3 presents the proposed passage ranking algorithms.

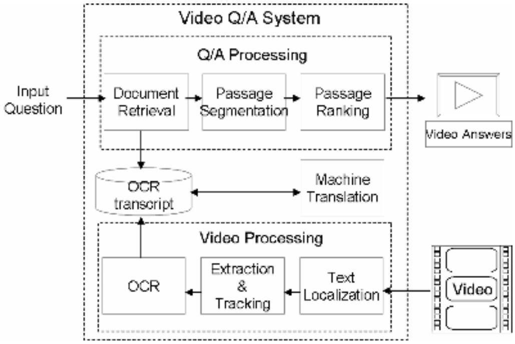


Fig. 1. Framework of video question answering

2.1 Video Processing

Our video processing takes a video and recognizes the closed captions as texts. An example of the input and output associated with the whole video processing component can be seen in Figure 2. Our videoOCR consists of three important steps: text localization, extraction&tracking, and OCR. The goal of text localization is to find the text area precisely. Here, we employ the edge-based filtering [1] [11] and slightly modify the coarse-to-fine top-down block segmentation methods [8] to locate text components in a frame. The former removes most non-edge areas with global and local thresholding strategy [5] while the latter incrementally segments and refines text blocks using horizontal and vertical projection profiles.

The target of extraction&tracking step is to integrate multi-frames and binarize the text pixels. As we know, the video consists of a sequence of image frames. A text component frequently appears more than one frames. To integrate them, we slightly modify Lin's approach [9] by counting the proportion of overlapping edge pixels between two frames. If the portion is above 70%, then the two frames were considered to contain the same text area and merge the two frames by averaging the gray-intensity for each pixel in the same text component. After multi-frame merging, the Lyu's text extraction algorithm [11] is used to binarize the text components. Unlike previous approaches [3] [9], this method does not need to assume the text is either bright or dark (but assume the text color is stable). At the end of this step, the output text components are prepared for OCR.

The aim of OCR is to identify the binarized text image to the ASCII text. In this paper, we re-implement a naïve OCR system based on nearest neighbor classification algorithms and clustering techniques [6] [18] [3]. We also adopted the word re-ranking methods (see [9], strategy 3) to improve the OCR errors.

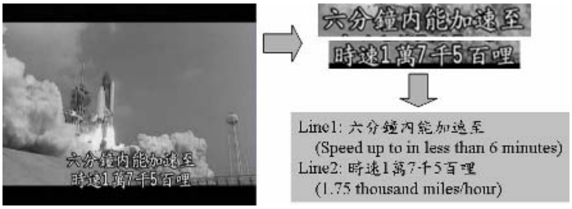


Fig. 2. Text extraction results of an input image

2.2 Document Processing and Passage Segmentation

Searching answers in the whole video collection is impractical since most of the words are irrelevant to the question. By means of text retrieval technology, the search space can be largely reduced and limited in a small number of relevant documents. The document retrieval methods have been developed well and successfully been applied for retrieving relevant documents for question answering [12] [14] [20]. We replicated the Okapi BM-25 [13], which is one of the most effective retrieval algorithms to find the related videoOCR documents for the input question. For the Chinese word indexing, we simply use the so-called “overlapping bigram”, i.e. two consecutive Chinese atomic characters that had been successfully applied for many Chinese document retrieval tasks [15].

Usually, words that occur in the same frame provide a sufficient and complet description. We thus consider that words that appear in the same frame as a sentence. The passage is then grouped with every 3 sentences and one previous sentence overlapping. An example of a sentence can be found in Fig. 2. The sentence of this frame is the cascading of the two text lines, i.e. “speed-up to 1.75 miles/hr in six minutes”

3 Passage Ranking Algorithm

The ranking model receives the segmented passages from previous steps and outputs the top-*N* passages to response the question. Tellex et al. [16] compared seven passage retrieval algorithms such as BM-25, density-based for TREC-Q/A task except for the two ad-hoc methods that needed either human-generated patterns or inference ontology which were unavailable. They indicated that the density-based methods achieved better performance than the other ranking models. In 2005 Cui et al. [4] showed that their fuzzy relation syntactic matching method outperformed the density methods by up to 78% relative performance. But the limitation is that it required a dependency parser, thesaurus, and training data. In many Asian languages like Chinese, Japanese, parsing is more difficult owing to it is necessary to resolve the word segmentation problem first. Besides, to develop a thesaurus and labeled training examples for Q/A is quite huge cost and time-consuming. Compared to Cui’s method, traditional term weighting models are much less cost, portable and practical.

In basic, traditional term weighting methods often give more weight on the passages that contain high frequent term match. Even density-based methods further take the word distribution into account, they still bias to high frequent terms if the passage tends to include abindant keywords rather than *N*-gram chunks. Usually, the *N*-gram information is much important than high-frequent unigram words. For example, the passage that contains three unigrams “optical”, “character“, “recognition” frequently receives similar score as the passage, which has the trigram “optical ∧ character ∧ recognition”. It is often the case that an *N*-gram chunk is much more unambiguous than its individual unigrams. Thus, we attempt to put emphasis on *N*-gram match and also take the density into account.

To efficiently find the match sequence, we design an algorithm that approximately discovers the optimal match sequence. At first, we note the following notations.

passage $\mathrm{P} = \mathrm{PW}_1, \mathrm{PW}_2, \ldots, \mathrm{PW}_N$
question $\mathrm{Q} = \mathrm{QW}_1, \mathrm{QW}_2, \ldots, \mathrm{QW}_M$
match sequence $\mathrm{S}_\mathrm{P} = \mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_N$

PW_i and QW_i are the *i*-th word in passage and question. Here, a word is viewed as the atomic Chinese character. The match sequence S_P represents the lexical match relation on the aspect of mapping each question word to the passage. The sequence is used to express whether the corresponding word in P is match(1) or nonmatch(0) a word in Q. From the point of probabilistic view, given a passage P we want to find the best-fit match sequence S_P that maximize Prob(S_P|P,Q). Prob(S_P|P,Q) can be considered as the generative probability of sequence S_P given P and Q. By applying Bayes rules, we have

$$\begin{aligned}\mathrm{Prob}(\mathrm{S_P} \mid \mathrm{P},\mathrm{Q}) &= \frac{\mathrm{Prob}(\mathrm{P},\mathrm{Q} \mid \mathrm{S_P})}{\mathrm{Prob}(\mathrm{P},\mathrm{Q})} \times \mathrm{Prob}(\mathrm{S_P}) \cong \mathrm{Prob}(\mathrm{P},\mathrm{Q} \mid \mathrm{S_P}) \times \mathrm{Prob}(\mathrm{S_P}) \\ &= \max_{\forall \mathrm{S_P}} \{\mathrm{Prob}(\mathrm{P},\mathrm{Q} \mid \mathrm{S_P}) \times \mathrm{Prob}(\mathrm{S_P})\}\end{aligned} \tag{1}$$

We skip Prob(P,Q) since it is equal for each match sequence. Prob($\mathrm{S_P}$) represents the generative probability of $\mathrm{S_P}$ among all possible state sequences. The above equation searches for the optimal sequence that generates P and Q with maximum probability. We further define the following equation to compute Prob($\mathrm{S_P} = \tilde{\mathrm{S}}_\mathrm{P}$) for a given state sequence $\mathrm{S_P}$.

$$\mathrm{Prob}(\mathrm{S_P} = \tilde{\mathrm{S}}_\mathrm{P}) = \frac{1}{Z(\mathrm{S_P})} \sum_{j=2}^{ngram\_cnt-1} \frac{|Sub_{j+1}|^{\alpha_1} + |Sub_j|^{\alpha_1}}{dist(Sub_j, Sub_{j+1})^{\alpha_2}} \times \frac{ngram\_cnt-2}{M} \tag{2}$$

where $Sub_j = \mathrm{PW}_m, \mathrm{PW}_{m+1}, \ldots., \mathrm{PW}_{m+n}$ subject to $Sub_j = \mathrm{QW}_{m'}, \mathrm{QW}_{m'+1}, \ldots., \mathrm{QW}_{m'+n}$ that is $\tilde{\mathrm{s}}_m = \tilde{\mathrm{s}}_{m+1} = \tilde{\mathrm{s}}_{m+2} = \ldots = \tilde{\mathrm{s}}_{m+n} = 1$

ngram_cnt represents the number of *N*-gram match for the match sequence $\tilde{\mathrm{S}}_\mathrm{P}$. Here we add two stable additional *N*-grams (*N*=1) at start-of-passage (SOP) and end-of-passage (EOP) which result least three matched *N*-grams for all match sequences. Sub_j is the *j*-th *N*-gram of $\tilde{\mathrm{S}}_\mathrm{P}$ which is an *N*-gram match between passage and question. If Sub_j starts from *m*-th word in P with *n* words length, which exact matches the equal length of *N*-gram question words in Q that starts from $\mathrm{QW}_{m'}$ to $\mathrm{QW}_{m'+\mathrm{n}}$. α_1 and α_2 are the parameters that controls the impact of subsequence length and the distance measurement. If we simply set α_1=1 and α_2=1, then equation (2) does only consider the number of N-grams match rather than the length of N-grams. We empirically set the two parameter as α_1=1.5 and α_2=0.5 which were found to be effective in our experiment. Z(S$_P$) is a normalizing constant determined by the requirement $\sum_{S_P} \text{Prob}(S_P) = 1$ that for all S$_P$:

$$Z(S_P) = \sum_{S_P} \sum_{j=2}^{ngram\_cnt-1} \frac{|Sub_{j+1}|^{\alpha_1} + |Sub_j|^{\alpha_1}}{dist(Sub_j, Sub_{j+1})^{\alpha_2}} \times \frac{ngram\_cnt-2}{M} \tag{3}$$

The normalization factor Z(S$_P$) of equation (2) does not effect the overall computing and it merely serves as a constant multiplier. By deduction of equation (1), the score of the given passage P is:

$$\begin{aligned} \text{Prob}(P, Q \mid S_P) &\cong \max_{S_P} \text{Prob}(S_P \mid P, Q) \times \text{Prob}(S_P) \cong \max_{S_P} \text{Prob}(S_P) \\ &\cong \max_{S_P} \sum_{j=2}^{ngram\_cnt-1} \frac{|Sub_{j+1}|^{\alpha_1} + |Sub_j|^{\alpha_1}}{dist(Sub_j, Sub_{j+1})^{\alpha_2}} \times \frac{ngram\_cnt-2}{M} \end{aligned} \tag{4}$$

The target of determining optimal match sequence $\tilde{S}_P$ is to find the best-fit $\tilde{S}_P$ that maximizes equation (4). However, there are 2^N possible sequences which are not able to find the optimum match sequence as statisfied with equation(4) in polynomial time. Therefore, we propose a Viterbi-like algorithm to approximately find the optimal match sequence using a dynamic programming technique (see Fig. 4.). Fig. 3. lists the preprocessing step of the proposed algorithm.

As shown in Fig. 3., the goal of algorithm 1 is to produce the element list, which stores all possible appearing locations of each question word in P. For question word QW$_i$, Loc^i_j indicates the j-th occurrence of QW$_i$ in P where $1 \le j \le L_i$, and a question word at most appears L_i times in P.

Algorithm 2 made use of a Viterbi-like method to compute the optimal score in forward direction, and track the path in the backward stage. The corresponding variable $\psi_t(\text{Loc}^t_j)$ records the node of the incoming arc that led to the most probable path. Using dynamic programming, we can calculate the most probable path through the whole trellis as outlined in algorithm 2. In Fig. 4, we define an equation to find the transition score between two elements in algorithm 2 (see equation (5)). Note that we give the discounting transition score[1] if the position $\text{Loc}^{t+1}_j < \text{Loc}^t_i$. In this way, the passages that preserve the word ordering sequence as the given question are given more weight. After the path tracing, Q*= $\{q^*_1, q^*_2, \ldots, q^*_M\}$ can be used to indicate the best-fit state sequence of S*= $\{s^*_1, s^*_2, \ldots, s^*_N\}$ by setting the corresponding location of q* as 1, while the others as 0. By applying equation (4), the S$_P$ score is obtained. Due to the length limitation, we left an example of estimating passage score using above algorithms at the web page[2].

$$a_{ij} = \frac{1}{y} \qquad \begin{cases} y = \text{Loc}^{t+1}_j - \text{Loc}^t_i & \text{if Loc}^{t+1}_j > \text{Loc}^t_i \\ y = |\text{Loc}^{t+1}_j - \text{Loc}^t_i|^2 & \text{otherwise} \end{cases} \tag{5}$$

[1] In this paper, we simply set the discounting factor as the square of the distance which resulted in satisfactory ranking performance.

[2] http://dblab87.csie.ncu.edu.tw/bcbb/tvqs/

```
Algorithm 1: Extracting_Possible_Match_Position (P, Q)
Input: given a passage P = $PW_1, PW_2, \ldots, PW_N$
        question Q = $QW_1, QW_2, \ldots, QW_M$
Output: Set of element list $\{Loc^1_1, Loc^1_2, \ldots, Loc^1_{L1},$
        $Loc^2_1, \ldots, Loc^2_{L2}, \ldots, Loc^M_1, \ldots, Loc^M_{LM}\}$
Algorithm:
        For (i := 1 ~ $N$) {
                $L_i$ := 0;
                For (j := 1 ~ $M$) {
                        If ($QW_i$ := $PW_j$) {
                                $Loc^i_{L_i}$ := j;
                        $L_i$ ++; }
        } }
```

Fig. 3. A preprocessing algorithm for finding the possible match positions

```
Algorithm 2: Finding-the-Optimal-Match-Sequence (set_of_element-list)
Input: Set of element list $\{Loc^1_1, Loc^1_2, \ldots, Loc^1_{L1},$
        $Loc^2_1, \ldots, Loc^2_{L2}, \ldots, Loc^M_1, \ldots, Loc^M_{LM}\}$
Output: Optimal path $Q$*= { $q^*_1, q^*_2, \ldots, q^*_M$ }
Algorithm:
    Initialization:  $\delta_1(Loc^1_i) = 1$   where $1 \le i \le L_1$
                     $\psi_1(Loc^1_i) = 0$
    Recursion:       $\delta_{t+1}(Loc^{t+1}_j) = \max_{1 \le i \le L_t} \{\delta_t(Loc^t_i) a_{ij}\}$
                     $\psi_{t+1}(Loc^{t+1}_j) = \arg\max_{1 \le i \le L_t} \{\delta_t(Loc^t_i) a_{ij}\}$
                     where $1 \le t \le M-1, 1 \le j \le L_{t+1}$
    Termination:     $q^*_M = \arg\max_{1 \le i \le L_M} \delta_M(Loc^M_i)$
    Backtracking:    $Q^* = \{q^*_1, q^*_2, \ldots, q^*_M\}$ so that $q^*_t = \psi_{t+1}(q^*_{t+1})$
```

Fig. 4. A Viterbi-like algorithm to find the optimal match sequence for a passage

Again, our method does do not simply consider the number of match words in a passage, instead, it seeks to find the match sequence that contains "dense" and "long" *N*-gram match relation between P and Q. However, the original density-based method does not tend to find the optimal match position for each word, rather it estimates the term distribution and weighted number of match words in the passage. But when there is only unigram match between P and Q, our method is then somewhat like traditional density-based approaches. The difference in this case relies in our method tries to find the optimal "one-to-one" match relation that leads to the densest distribution rather than "one-to-all".

4 Experimental Result

4.1 Dataset and Evaluations

The testing video corpus mainly collected from the 93 Discovery films. Table 1 sumarizes the characteristics of this corpus. In comparison with previous videoQ/A studies [2] [9] [18] [19] [21] which made use of less than 6 films and less than 40 questions, the used video collection and testing questions are substantially larger. The video data had been converted into OCR document through the video processing (see section 3). Then, we found that video OCR method recognized 49951 sentences and 677060 single Chinese characters. Averagely, a video contains 537 sentences and 7280 Chinese characters.

We collect 150 actual user's questions to evaluate the overall video Q/A performance. By following the answer assessment process as TREC-Q/A task [17], each question was judged by two assessors and differences were labeled. A domain expert reviewed each ambiguous label to determine or correct the final answer. If the system returns the correct answer frames, then it is judged as right answer. In this paper, we use MRR score (Voorhees, 2000) which is widely used for evaluation to measure the passage ranking performance, while the recall and precision rates are used to evaluate the accuracy.

Table 1. Statistics of the Discovery programs

Number of videos	93
Number of recognized sentences	49951
Number of recognized Chinese characters	677060
Total video data size	45.2GB

4.2 Results

At the beginning, we evaluate our videoOCR method in terms of text localization and overall OCR recognition rates. The Discovery collection is a large video dataset, which makes it difficult to comprehensively test on it. Rather, we only examine our method on a small sampled set of the vides that collected from 30 short clips. Our videoOCR sampled two frames per second with 352x240 resolution for the MPEG-1 Discovery movies. Totally there were 1684 image frames derived from the 30 clips which contained 2195 text areas. Table 2 lists the experimental results on text localization (Table 2(a)) and the overall videoOCR (Table 2(b)).

For the overall videoQ/A evaluation, we compare our ranking algorithm with TFIDF (term frequency multiplies inverse document frequency), BM-25 [13], and density-based [7] approaches. These ranking models received the same segmented passages, and retrieved top-ranked passages with their own methods. Empirically, we found that on averagely, the retrieved passage contains 48.78 Chinese words that represent a very short but more complete fragment than merely retunring an answer phrase. Table 3 summarizes the overall videoQ/A results using different ranking models.

As seeing in Table 3, our method produced 0.572 MRR(top5) score which outperformed the TFIDF, BM-25, and density-based approaches. In comparison to the second best method (TFIDF), the proposed ranking algorithm is relatively 6.31%

better in MRR(top5) score, and 5.52%, 5.75% better in terms of recall and precision rates. Compared to the BM-25 model, our method is 19.16%, 16.02%, and 15.72% better in relatively MRR(top5), recall, precision rates.

Table 2(a). Text localization performance with pixel and area based evaluations

Recall (Pixel-based)	Precision (Pixel-based)	Recall (Area-based)	Precision (Area-based)
92.97%	95.06%	98.93%	97.63%

Table 2(b). Overall videoOCR performance

Recall	Precision	F1-measure
86.51%	83.05%	84.74%

Table 3. System performance for different passage ranking algorithms

	TFIDF	BM-25	Density-based	**Our method**
MRR (Top1)	0.479	0.413	0.280	**0.506**
MRR (Top5)	0.538	0.480	0.370	**0.572**
Recall (Top5)	0.597	0.543	0.472	**0.630**
Precision (Top5)	0.174	0.159	0.137	**0.184**

5 Conclusion

Usually, it is necessary to integrate with abundant external knowledge for answering. This paper proposes a light-weight and multilingual portable video Q/A system that extend the text Q/A method to multimedia. The system returns the retrieved passages with corresponding video clips in response to the question. The experiments showed that the proposed method outperforms the TFIDF, TF, BM-25, and density-based approaches in terms of MRR score. In the future, we plan to adopt well known speech recognition techniques to enhance the system performance.

References

1. Cai, M., Song, J., and Lyu, M. R. A new approach for video text detection. In Proceedings of International Conference on Image Processing, pages 117-120, 2002.
2. Cao, J., and Nunamaker J. F. Question answering on lecture videos: a multifaceted approach, International Conference on Digital Libraries, pages 214 – 215, 2004.
3. Chang, F., Chen, G. C., Lin, C. C., and Lin, W. H. Caption analysis and recognition for building video indexing systems. ACM Multimedia systems, 10(4): 344-355, 2005.
4. Cui, H., Sun, R., Li, K., Kan, M., and Chua, T. Question answering passage retrieval using dependency relations. In Proceedings of the 28th ACM SIGIR Conference on Research and Development in Information Retrieval, pages 400-407, 2005.
5. Fan, J., Yau, D. K. Y., Elmagarmid, A. K., and Aref, W. G. Automatic image segmentation by integrating color-edge extraction and seeded region growing. IEEE Trans. On Image Processing, 10(10): 1454-1464, 2001.

6. Hong, T., Lam, S. W., Hull, J. J., and Srihari, S. N. The design of a nearest-neighbor classifier and its use for japanese character recognition. In Proceedings of Third International Conference on Document Analysis and Recognition, pages 270-291, 1995.
7. Lee, G. G., Seo, J. Y., Lee, S. W., Jung, H. M., Cho, B. H., Lee, C. K., Kwak, B. K., Cha, J. W., Kim, D. S., An, J. H., and Kim, H. S. SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP. In Proceedings of the 10th Text Retrieval Conference, pages 437-446, 2001.
8. Lienhart, R. and Wernicke, A. Localizing and segmenting text in images and videos. IEEE Trans. Circuits and Systems for Video Technology, 12(4): 243-255, 2002.
9. Lin, C. J., Liu, C. C., and Chen, H. H. A simple method for Chinese video OCR and its application to question answering. Computational linguistics and Chinese language processing, 6(2): 11-30, 2001.
10. Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., Karger, D. R. What makes a good answer? the role of context in question answering. In Proceedings of the 9th international conference on human-computer interaction (INTERACT), page 25-32, 2003.
11. Lyu, M. R., Song, J., and Cai, M. A comprehensive method for multilingual video text detection, localization, and extraction. IEEE Trans. Circuits and Systems for Video Technology, 15(2): 243-255, 2005.
12. Pasca, M., and Harabagiu, S. High-performance question answering. In Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval, pages 366-374, 2001.
13. Robertson, E., Walker, S., and Beaulieu, M. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. In Proceedings of the 7th Text Retrieval Conference, 1998.
14. Rus, V., and Moldovan, D. High precision logic form transformation. International Journal on Artificial Intelligence Tools, 11(3): 437-454, 2002
15. Savoy, J. Comparative study on monolingual and multilingual search models for use with Asian languages. ACM transactions on Asian language information processing (TALIP), 4(2): 163-189, 2005.
16. Tellex, S., Katz, B., Lin, J. J., Fernandes, A., and Marton, G. Quantitative evaluation of passage retrieval algorithms for question answering. In Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval, pages 41-47, 2003.
17. Voorhees, E. M. Overview of the TREC 2001 question answering track. In Proceedings of the 10th Text Retrieval Conference , pages 42-52, 2001.
18. Wu, Y. C., Lee, Y. S., Chang, C. H. CLVQ: Cross-language video question/answering system. In Proceedings of 6th IEEE International Symposium on Multimedia Software Engineering, pages 294-301, 2004.
19. Yang, H., Chaison, L., Zhao, Y., Neo, S. Y., and Chua, T. S. VideoQA: Question answering on news video. In Proceedings of the 11th ACM International Conference on Multimedia, pages 632-641, 2003a.
20. Yang, H., Chua, T. S., Wang, S. G., and Koh, C. K. Structural use of external knowledge for event-based open domain question answering. In Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval, pages 33-40, 2003b.
21. Zhang, D., and Nunamaker, J. A natural language approach to content-based video indexing and retrieval for interactive E-learning. IEEE Transactions on Multimedia, 6(3): 450-458, 2004.

Fusion of Luma and Chroma GMMs for HMM-Based Object Detection

Wen-Hao Wang and Ruei-Cheng Wu

Industrial Technology Research Institute, Taiwan
{devin, allen}@itri.org.tw

Abstract. A spatial-temporal method is proposed for video object detection. The first stage is temporal segmentation which is comprised of GMM on luma and GMM on chroma. An efficient fusion method is proposed to combine the two GMM segmentation results such that the object mask can be improved to some extent. In the second stage, the mask produced in the first stage is statistically analyzed in spatial domain. HMM is employed to refine the segmentation result by estimating the foreground-background state such that the false detection in foreground and background area can be decreased and lead to robust and satisfactory detection results.

1 Introduction

Object detection is a very important task in many video applications, *e.g.* computer vision and video surveillance *etc*. In general, object detection is the main factor to the success of a vision system. Thus, there are many researches regarding the design of robust object detection.

In sequential video processing, GMM [6] (Gaussian mixture model, *i.e.* Stauffer-Grimson algorithm [1]) is usually used to model each pixel (or region) in order to adapt the background model to the changing illumination. Pixel values that do not fit the model will be considered as foreground. A practical implementation using GMM is the human action recognition system [7] which employed GMM for real-time moving object detection. The detail discussion of the implementation of Stauffer-Grimson algorithm can be found in [8].

HMM (Hidden Markov Model) is useful for modeling non-stationary process considering that a temporal continuity constraint can be imposed on consecutive pixels' intensities; *i.e.*, if a pixel is detected as part of foreground (object), it is expected to remain part of the foreground for a period of time. The advantages of HMM revealed in [2] are as follows: (1) Selection of training data is not required (unsupervised); (2) Different hidden states allow the learning of statistical characteristics of foreground and background from a mixed sequence of foreground symbol and background symbol, *e.g.* in the proposed method, the 1D representation (described in Sec. 2.3) of the binary segmentation object mask. In [3], there are three states for each pixel: foreground state, background state and shadow state; however, HMM for each pixel is inclined to be slow. In addition, HMM will cost considerable amount of time for estimating HMM parameters owing to the fact that Baum-Welch algorithm [5] is an iterative likelihood maximization method and tends to be time-consuming. In order to

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 573 – 581, 2006.

be used in practical video surveillance application, the proposed method also presents a rapid way of estimating HMM parameters instead of using Baum-Welch algorithm.

The paper is organized as follows: The proposed object detection method is described in Sec. 2. Sec. 3 demonstrates the experimental results. Finally, a brief conclusion is given in Sec. 4.

2 Proposed Method

Fig. 1 illustrates the proposed method for object detection. $\psi_C(t)$ denotes chroma (*i.e.* hue and saturation) of an input image $\psi(t)$ at time *t*. $\psi_I(t)$ represents the luma (intensity) of the input image. In this paper, GMM [1] is used to model the background statistics. In order to not to be affected by the shadow, the input color space (RGB) is transformed to HSI space. 'HS' are hue and saturation which are not sensitive to the appearance of shadow. 'I' denotes 'intensity' which can lead to better segmentation than HS; however, it is susceptible to shadow of objects. After the processing of GMM, the preliminary segmented object mask of ψ_C (ψ_I) is Ω_C (Ω_I) as illustrated in the block diagram of Fig. 1. In the block diagram, all masks are represented by

$$\Omega(x,y,t) = \{ \begin{matrix} 1 & \psi(x,y) \text{ is foreground} \\ 0 & \psi(x,y) \text{ is background} \end{matrix} \tag{1}$$

Sec. 2.1 will introduce the principles of GMM and the shadow issue.

"Mask Fusion" is designed to combine the positive effects of segmentation results of chroma and luma. In general, HS part of an image cannot lead to satisfactory segmentation result in background area but it can avoid the shadow. On the other hand, luma can lead to better background segmentation than chroma; however, shadow will also be segmented as part of foreground. Thus, Sec. 2.2 will discuss the fusion approach to combing the two segmentation results.

In Sec. 2.3, HMM (Hidden Markov Model) is employed to model the background in spatial domain in contrast to that GMM is used to model the background in temporal domain for each pixel.

2.1 Gaussian Mixture Model

All pixels in a specific 'pixel position' (x, y) of an image sequence over time is considered as "random process" or "pixel process" [1] as expressed as follows:

$$\Delta = \{X_1,...,X_T\} = \{\psi(x,y,t) \mid t \in (1,T)\} \tag{2}$$

For on-line applications, the set $\Delta$of recent collection of observations $\{X_1,...,X_t\}$ can be modeled by K Gaussian distributions as

$$p(x \mid \Delta, \alpha, \beta) = \sum_{j=1}^{K} \omega_j \Phi(x, \mu_j, \sigma_j^2 I) \tag{3}$$

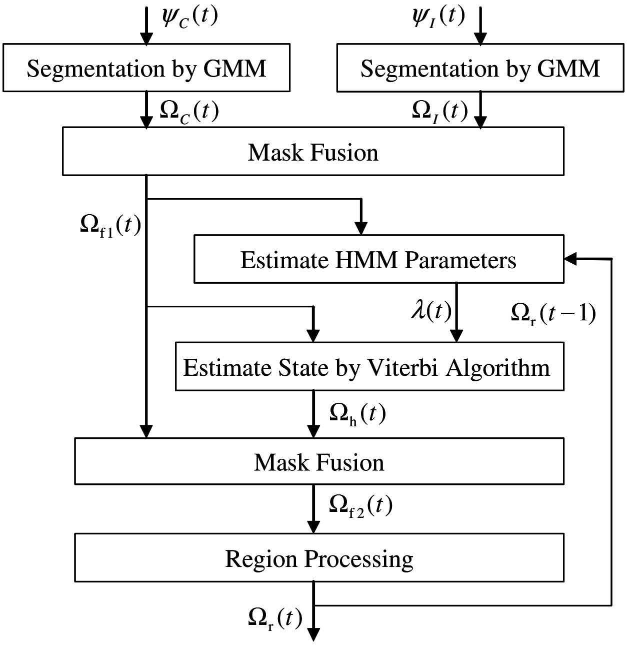


Fig. 1. Block diagram of the proposed object detection method

condition on observation set Δ, foreground α, and background β. ω_j is the weight of the j(th) Gaussian distribution Φ. I is identity matrix. Finally, the background can be approximated by

$$p(x \mid \Delta, \beta) = \sum_{j=1}^{B} \omega_j \Phi(x, \mu_j, \sigma_j^2 I) \tag{4}$$

i.e., the background is modeled by the first B Gaussian distributions whose (ω/σ) ratios are the first B largest values [1].

GMM modeling is employed as the first step of the proposed object detection method. It is to model the background by chroma and luma, respectively. In conventional method, RGB color space is used to model the input image sequence. However, shadow will be extracted as part of object mask when using RGB color space. HSI color space is used in segmentation [4] and tends to be shadow-removable. However, the results are not satisfactory due to the fragmented segmentation results by using hue and saturation. In order to achieve both 'shadow-rejection' and 'segmentation stability over time,' GMM is employed on chroma (hue and saturation) and luma (intensity) separately. The segmentation result by chroma (luma) is denoted as Ω_C (Ω_I) as illustrated in Fig. 1.

2.2 Fusion of Object Mask

In this section, two feature values will be computed. Firstly, let $\Gamma_1(\mathrm{x},\mathrm{y})$ denote the statistics (foreground appearance) of the 3x3 block centering at $\Omega_{\mathrm{C}}(\mathrm{x},\mathrm{y})$.

$$\Gamma_1(\mathrm{x},\mathrm{y}) = \sum_{i=-1,j=-1}^{i=1,j=1} \Omega_C(x+i, y+j) \tag{5}$$

Secondly, let $\Gamma_2(\mathrm{x},\mathrm{y})$ denote the accumulated support resulted from chroma and luma.

$$\Gamma_2(\mathrm{x},\mathrm{y}) = \sum_{i=-1,j=-1}^{i=1,j=1} \Omega_C(x+i, y+j) + \Omega_{\mathrm{I}}(x+i, y+j) \tag{6}$$

The principle of fusion of Ω_{C} and Ω_{I} is formulated as follows:

Criterion 1: if $\Gamma_1(\mathrm{x},\mathrm{y}) \geq \rho_1$, $\Omega_{\mathrm{f1}}(\mathrm{x},\mathrm{y}) = 1$, otherwise, $\Omega_{\mathrm{f1}}(\mathrm{x},\mathrm{y}) = 0$ (7)

Ω_{f1} is the mask-fusion result. By using this criterion, the resultant binary object mask will not include shadow part because only chroma information is used for judgment. However, this mask is not satisfactory because of poor performance of chroma. By using the following second criterion, the binary mask Ω_{f1} can be further improved without being affected by shadow.

Criterion 2: if $\Gamma_2(\mathrm{x},\mathrm{y}) \geq \rho_2$, $\Omega_{\mathrm{f1}}(\mathrm{x},\mathrm{y}) = 1$, otherwise, $\Omega_{\mathrm{f1}}(\mathrm{x},\mathrm{y}) = 0$. (8)

ρ_1 (*e.g.* 2) and ρ_2 (*e.g.* 8) are thresholds. By means of this two criteria, Ω_{f1} can reserve the merit of the result using chroma (*i.e.* avoiding shadow) and the merit of the result using luma (*i.e.* stability of segmentation).

2.3 Refine Object Mask by HMM

The 1D signal representation of the 2D object mask can be considered as a non-stationary signal process which is composed of several state-dependent subprocesses. For example, in object detection problem, there are two states: background state S_1 and foreground state S_2 as illustrated in Fig. 2. Each subprocess is a Markovian chain with stationary statistics. Thus, a mask (represented as 1D sequence) can be modeled by HMM. An HMM model is abbreviated as

$$\mathrm{H} := (N, M, A, \pi, P_1, P_2). \tag{9}$$

N is the number of states. *M* is the number of symbols. In the proposed design, *N* = 2 (S_1 : background state, and S_2: foreground state) and *M* = 2 (background symbol and foreground symbol). *A* is the state transition probability matrix.

$A = \{a_{ij}, i, j = 1,...N\}$, a_{ij} is the transition probability from state *i* to state *j*. In the proposed method, the state transition probability is initialized as: $a_{11} = 0.9, a_{12} = 0.1, a_{21} = 0.1, a_{22} = 0.9$. The state diagram is illustrated in Fig. 2.

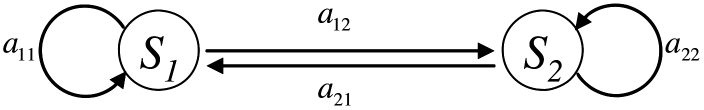


Fig. 2. The HMM model of background and foreground

$\pi = \{\pi_1,...,\pi_N\}$, π_i is the initial state probability of sate *i*. P_1 and P_2 is the PDF (probability density function) of state 1 and state 2, respectively. $P_1(\mathrm{x} = \alpha)$ is the probability that foreground symbol occurs during the background situation. $P_1(\mathrm{x} = \beta)$ is the probability that background symbol occurs during the background situation. $P_2(\mathrm{x} = \alpha)$ is the probability that foreground symbol occurs during the foreground situation. $P_2(\mathrm{x} = \beta)$ is the probability that background symbol occurs during the foreground situation. In Fig. 1, λ denotes the set of HMM parameters in the model H.

Fig. 3 shows the generic flow of HMM procedure. The first step is to initialize the HMM parameters. The second step is to estimate and update the HMM parameters by Baum-Welch algorithm. The last step of the procedure is to estimate the state of the input sequence by means of Viterbi algorithm. In the proposed approach, the estimated state for each pixel is background state or foreground state.

The proposed method of re-estimating (refining) the background mask is formulated as a HMM training problem to obtain the model parameters. As shown in Fig. 3, Baum-Welch method [5] is used for training parameters of HMM. It is an iterative likelihood maximization method. By means of Baum-Welch algorithm, the transition matrix *A*, state probability π_i of each state, and the PDF of each state P_{i} can be trained and updated according to previous samples.

At the last step of HMM procedure, Viterbi decoding algorithm is used to obtain the maximum a posterior (MAP) estimate of the states of an unlabelled observation sequence (*i.e.* the 1D representation of the object mask).

For practical implementation of HMM, Baum-Welch algorithm is not employed in the HMM procedure because it tends to be time-consuming. In order to estimate the HMM parameters in a faster way, the training stage of HMM is modified as follows.

Let $\Omega(\mathrm{t})$ denote the binary mask excluding the foreground part of previous final object mask $\Omega_{\mathrm{r}}(\mathrm{t}-1)$:

$$\Omega(\mathrm{t}) = \overline{\Omega_{\mathrm{r}}(\mathrm{t}-1)}\ \mathrm{AND}\ \Omega_{\mathrm{f1}}(\mathrm{t}). \tag{10}$$

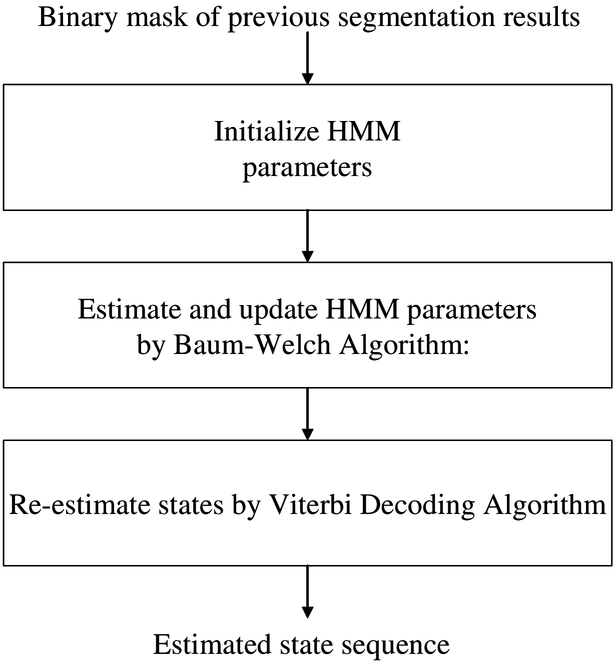

Binary mask of previous segmentation results
Initialize HMM parameters
Estimate and update HMM parameters by Baum-Welch Algorithm:
Re-estimate states by Viterbi Decoding Algorithm
Estimated state sequence

Fig. 3. HMM procedure

Let ξ denote the occupy-ratio of foreground symbol in $\Omega(t)$. Thus, the probability of foreground in background state can be approximated as

$$P_1(\mathrm{x}=\alpha)=\xi\,. \tag{11}$$

Therefore, the probability of background symbol in background state is as follows:

$$P_1(\mathrm{x}=\beta)=1-P_1(\mathrm{x}=\alpha)\,. \tag{12}$$

After updating the HMM parameters, the Viterbi algorithm is used to estimate the state of each element of $\Omega_{f1}(x,y,t)$. In other words, the statistical model of the background is estimated; if part (fusion of background symbol and foreground symbol) of $\Omega_{f1}(t)$ fit the background statistics, the part will be recognized as background. Therefore, the original object mask $\Omega_{f1}(t)$ can then be refined and result in a better object mask $\Omega_h(t)$.

In the proposed method, the HMM procedure can be performed in different scale options. The above processing is scale = 1, *i.e.* the original resolution of mask is used for processing. In case of scale=2 for HMM procedure, $\Omega(\mathrm{t})$ will be down-sampled to $\Omega'(\mathrm{t})$ and $\Omega'(\mathrm{t})$ will be used instead of $\Omega(\mathrm{t})$ for estimating HMM parameters. In this case, the refined state sequence is denoted as $\Omega'_h(t)$ which must be up-sampled

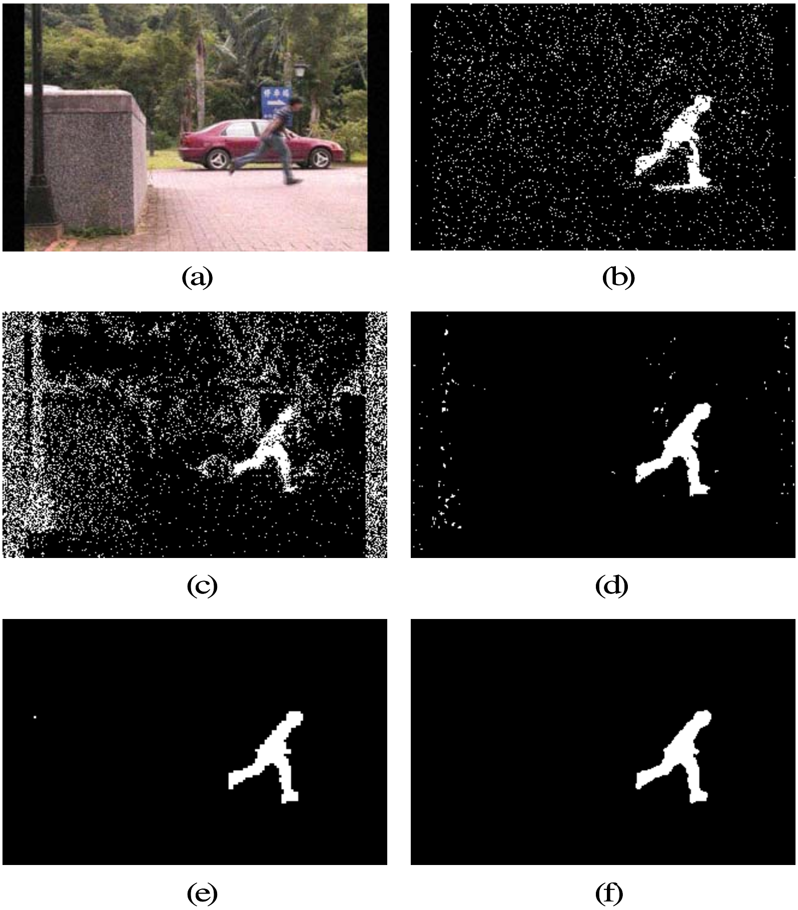


Fig. 4. (a) Frame 100, (b) Ω_C , (c) Ω_I , (d) Ω_{f1} , (e) Ω''_h (scale = 2), (f) Ω_r

to obtain the object mask $\Omega''_h(t)$ (with original size) in this HMM procedure. According to experimental results, HMM of scale=2 will lead to more robust object mask. In the final stage of Fig. 1, "Regions processing," connected component analysis (CCA) [9] is used to as the post-processing to refine the object mask $\Omega_{f2}(t)$. The final result is denoted as $\Omega_r(t)$.

3 Experimental Results

On considering that the area of the background is statistically larger than area of foreground in general case, the initial probability for background state is set as $\pi_1 = 0.9$ and the initial probability for foreground state is set as $\pi_2 = 0.1$.

In the experiments, the first twenty-three frames are captured in case of no objects. Fig. 4 shows the experimental results of frame 100. Fig. 4 (a) is the frame 100 with zero-mean additive Gaussian noise. Fig. 4 (b) is the GMM segmentation results of luma (intensity). It is noted that that the shadow cannot be avoided. This is inevitable when using luma for segmentation. On the contrary, the object mask will not contain the shadow part by using chroma (hue and saturation) for segmentation as shown in Fig. 4 (c). As mentioned in Sec. 2, the performance is not satisfactory. The only positive result of chroma is shadow-rejection. Fig. 4 (d) is the fusion result of (b) and (c) by means of the proposed fusion method described in Sec. 2.2. It is prominent that the resultant mask preserves the fine parts of (b) and (c); the foreground part becomes more stable and the background part become clearer. It is obvious that a complete object mask can be almost extracted at this step.

Fig. 4 (e) shows the refined results by means of the proposed HMM approach which lead to the estimated state results (white dot denotes foreground state and black dot represents background state) by means of the proposed fusion method.

At this experiments, HMM procedure is performed under scale = 2 which will result in more distinguishable object mask than scale=1. Fig. 4 (f) is the result of mask fusion and region processing (the last two steps of Fig. 1). It can be seen that “mask fusion” can also achieve smoothing the contour of the estimated mask (states) by HMM especially when scale = 2.

4 Conclusions

An improved spatial-temporal object detection method was presented in this paper. The detection is divided into two stages. The first stage is to extract object by applying GMM on luma and chroma separately and a criteria-based fusion method is designed to combine the two segmentation results. In the second stage, HMM procedure is employed to estimate the probability of the occurrences of foreground and background symbols in background state. Instead of conventional time-consuming Baum-Welch algorithm, the proposed method of updating HMM parameters is employed to speed up the process of estimating HMM parameters. With the updated HMM parameters, the foreground-background states are re-estimated by Viterbi decoding algorithm. According to the experimental results, robust object detection can be effected by means of the proposed method of HMM-based object detection with the fusion of luma and chroma GMM results.

Acknowledgements

This work was supported by ITRI project code 5301XS3L10.

References

1. Stauffer, C., Grimson, W.E.L.: Adaptive Background Mixture Models for Real-Time Tracking. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, (1999) 23-25
2. Rittscher, J., Kato, J., Joga S., Blake A.: A Probabilistic Background Model for Tracking

3. Kato, J., Joga, S., Rittscher, J., Blake, A.: An HMM-Based Segmentation Method for Traffic Monitoring Movies. IEEE Trans. PAMI, Vol. 24, No. 9 (2002) 1291-1296
4. Li, N., Bu, J., Chen, C.: Real-Time Video Object Segmentation Using HSV Space. ICIP, (2002) II-85-II-88
5. Junag, B.-H., Rabiner, L.R.: Mixture Autoregressive Hidden Markov Models for Speech Signals. IEEE Trans. Acoustics, Speech, and Signal Processing, (1985) 1404-1413
6. Zivkovic, Z.: Improved Adaptive Gaussian Mixture Model for Background Subtraction. ICPR, Vol.2, (2004) 28-31
7. Dedeoglu, Y.: Human Action Recognition Using Gaussian Mixture Model Based Background Segmentation. Machine Learning Workshop, Bilkent University (2005)
8. Power, O. W.: Understanding Background Mixture Models for Foreground Segmentation. Proceedings of Image and Vision Computing New Zealand, (2002) 267-271
9. Horn, K.P.: Robot Vision, MIT Press, (1986) 299-333

Undistorted Projection onto Dynamic Surface

Hanhoon Park, Moon-Hyun Lee, Byung-Kuk Seo, and Jong-Il Park*

Division of Electrical and Computer Engineering, Hanyang University, Seoul, Korea
Phone: +82-2-2220-0368; fax: +82-2-2299-7820
hanuni@mr.hanyang.ac.kr, fly4moon@teramail.com,
nwseoweb@mr.hanyang.ac.kr, jipark@hanyang.ac.kr
http://mr.hanyang.ac.kr

Abstract. For projector-based augmented reality systems, geometric correction is a crucial function. There have been many researches on the geometric correction in the literature. However, most of them focused only on static projection surfaces and could not give us a solution for dynamic surfaces (with varying geometry in time). In this paper, we aim at providing a simple and robust framework for projecting augmented reality images onto dynamic surfaces without image distortion. For this purpose, a new technique for embedding pattern images into the augmented reality images, which allows simultaneous display and correction, is proposed and its validity is shown in experimental results.

Keywords: Projector-camera system, augmented reality, geometric correction, dynamic surface, pattern embedding.

1 Introduction

Augmented reality is a technology in which a user's view of the real world is enhanced or augmented with additional (graphical) information, such as virtual objects, virtual illumination, explanatory text, etc., generated by a computer. Conventionally, monitor, video see-through and optical see-through head-mounted display (HMD) have been the main displays for augmented reality. In recent years, projectors become widespread due to their increasing capabilities and declining cost. Being able to displaying images with very high spatial resolution virtually anywhere is an interesting feature that cannot be provided by desktop screens. Moreover, the size of projectors is getting smaller and handheld projectors are emerging [8]. Thanks to these advances, projector-based augmented reality is getting more attention and applied to a variety of applications recently [2, 9, 12]. However, to make projector-based augmented reality feasible, lots of techniques such as geometric correction and radiometric compensation are required [1]. Among them, we focus on the geometric correction methods in this paper. Actually, there have been many researches on the geometric correction in the literature [1, 7, 9, 10]. However, most of them were available only on static projection surfaces while we are interested in dynamic projection surfaces.

[*] Corresponding author.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 582 – 590, 2006.

In general, surface geometry should be recovered for geometric correction. Most of projector-based augmented reality systems use structured light techniques [1, 2, 11] to recover surface geometry. That is, after projecting a pattern image with known geometry, the surface geometry is computed from the pattern image captured by a camera. To recover the geometry of dynamically changing surface with this mechanism, the pattern image should be projected continuously onto the surface. In this case, the projected pattern may prevent users from seeing the augmented reality image. Therefore, the structured light techniques cannot be directly applied to recovering dynamic surface geometry in projector-based augmented reality systems.

A good way of recovering the dynamic surface geometry in projector-based augmented reality systems would be to project a pattern image in such a way that the pattern image is not visible to users. In this paper, pre-determined pattern images are embedded into a sequence of augmented reality images by adding a certain value (brightness or color) to the pixels of odd frames of the sequence while subtracting the value to the pixels of even frames of the sequence. The embedded pattern image would be invisible externally but detectable using a simple image processing algorithm (e.g. image subtraction). Therefore, the invisible pattern image can be used for recovering the dynamic surface geometry.

1.1 Related Works

There are some researches associated with real-time surface geometry recovery [3, 13, 14]. In the field of projector-based augmented reality, Yasumuro et al. [14] used an additional infrared projector to project near-infrared patterns which are invisible to human eyes. Their system can project augmented reality images on dynamic human body without distortion and distraction at the expense of complicated system using at least two different types of projector. Cotting et al. [3] measured the mirror flip (on/off) sequences a Digital Light Processing (DLP) projector for RGB values using a photo transistor and a digital oscilloscope and imperceptibly embedded arbitrary binary patterns into augmented reality images by mapping the original augmented reality image values to the nearest values in such a way that the mirror flips are adjusted in the period in which no mirror flips occur. However, sophisticated control of camera shuttering for detecting the short-term patterns is required. In the field of 3D video capturing, Vieira et al. [13] alternatively projected color complementary pattern images onto a scene for obtaining the geometric and photometric information of the scene. Their system is able to generate 3-D video in real-time but not likely to be applied to augmented reality.

1.2 Contribution of This Paper

This paper provides a solution for projecting image onto *dynamic* surface without image distortion, which has been less investigated in the projector-based augmented reality society. The method works with any types of projectors while the previous methods suppose to use a particular (usually expensive) projector such as IR projector and DLP projector.

This paper proposes a new pattern embedding technique for *dynamic geometric correction* in projector-based augmented reality. Our method is inspired by the complementary patterns proposed by Vieira et al. [13]. The main difference lies in the fact that the pattern images in this paper are modulated into the AR images by adding/subtracting a certain value (brightness or color) to the pixels of the augmented reality image.

2 Estimation of Dynamic Surface Geometry

For projecting an image onto non-planar projection surface without image distortion, the surface geometry should be known [7]. Especially, when the geometry of the projection surface varies in time, the process of estimating the surface geometry should be performed in real-time. The previous structured light techniques [11] are good for the purpose of real-time geometry estimation. However, they cannot be directly applied to augmented reality because the structured light (pattern image) projected together with augmented reality image may visually distort the augmented reality image. Thus, we propose a novel method for recovering the dynamic surface geometry. In our method, a predefined pattern image is embedded into augmented reality image such that the pattern image is not noticeable subjectively but detectable using a simple image processing algorithm. The detected pattern image can be directly used for recovering the dynamic surface geometry.

2.1 Embedding and Detecting Pattern Images

Given a sequence of augmented reality images, we can embed a pattern image by adding and subtracting a certain value (brightness or color) to the pixels of odd and even frames of the sequence respectively (see Fig. 1). The pattern image would be externally invisible when the frame rate of the sequence is doubled because the increase and the decrease in the pixel value are offset by the characteristic of the human vision system (persistence of vision).

Note that the degree of the invisibility varies according to the magnitude or the kind of the variation of the pixels. In general, a large variation makes the pattern image robust to noise but noticeable while a low variation makes the pattern image completely invisible but weak to noise. Humans are sensitive to the variation of brightness but less sensitive to that of color [6]. Especially, humans are least sensitive to the variation of blue color. Experimentally, the variation of 50~60 in the blue channel is desirable. If the projection surface geometry is changed in time, the pattern image is not completely invisible any more. However, if the change is slow or at long intervals, the pattern image would be unnoticeable and our method is still useful. The difference between the odd and even frames may also affect the visibility of the pattern image. However, the variation between the successive frames is usually very small and thus ignorable.

Subtraction operation can be used for extracting the embedded pattern images from the odd and even frames as shown in Fig. 2. After subtraction, the resulting image could be noisy so the median filtering is applied to the image. In our experimental environment, these simple operations resulted in the pattern images with high quality. More sophisticated algorithms may be employed under ill-conditioned experimental environments.

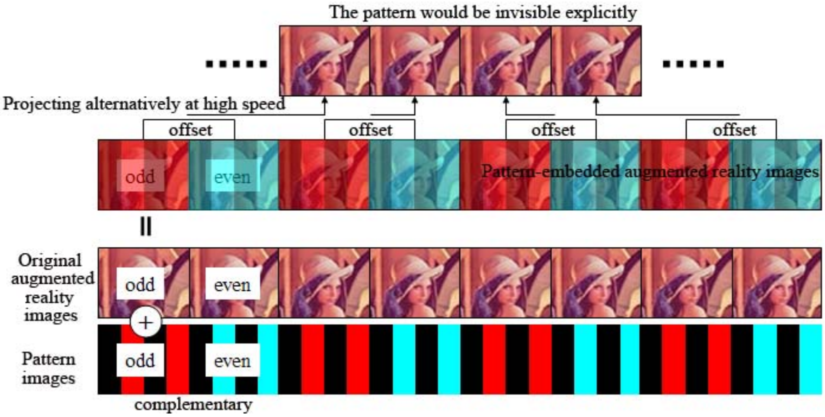

The pattern would be invisible explicitly
Projecting alternatively at high speed
offset
offset
offset
offset
odd
even
Pattern-embedded augmented reality images
Original augmented reality images
odd
even
+
Pattern images
odd
even
complementary

Fig. 1. Pattern embedding. The complementary pattern images are embedded into the odd/even frames. If the pattern-embedded odd and even frames are projected alternatively at a doubled frame rate, the complementary pattern images would be compensated subjectively and thus invisible.

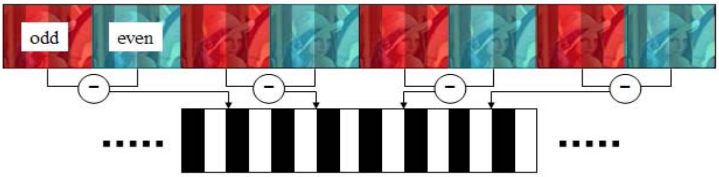


Fig. 2. Pattern detection. The embedded pattern images are extracted by the absolute difference between the corresponding odd and even frames.

2.2 Surface Geometry Recovery

Once the projectors and the camera are calibrated using a variant of the Zhang's calibration method [13, 14], the projection surface geometry is recovered using a simple binary-code method based on temporal coding [11] and a linear triangulation method [5]. A set of patterns are successively projected onto the surface and imaged by a camera. The codeword for a given pixel is usually formed by the sequence of illumination value for that pixel across the projected patterns. The correspondences between the projector input images and the camera images are found based on the codeword [11]. Finally, their 3D position is computed by applying triangulation to the corresponding pixels in the projector input image and the camera image. The surface is represented with triangular meshes generated by using the recovered 3D points.

The temporal coding method requires several pattern images to be projected, which implicitly supposes that the surface geometry is static. However, even if the surface geometry varies in time, this mechanism is still useful if the variation is not large [11]. Therefore, it is assumed that the surface geometry varies slowly in time in this paper. This limitation would be mitigated if another method which can recover the surface geometry using a single pattern image is employed at the risk of being unreliable [11].

3 Geometric Correction

Because the projection surface is represented piece-wise planar (with triangle meshes) as aforementioned, the geometric relationship between projectors, camera, and the projection surfaces can be explained by homographies [1, 7, 12]. The Discrete Linear Transform (DLT) algorithm can be used to calculate homographies between the projectors and the camera (or user's viewpoint) via the triangle meshes. More sophisticated algorithms can be used for obtaining a reliable solution [10]. If the projectors, cameras are calibrated and the projection surface geometry is known, it is possible to warp the projection image using the homographies to be undistorted in an arbitrary viewpoint [7].

An arbitrary viewpoint image m can be synthesized by projecting the points M on the projection surface onto the viewpoint image plane as

$$m = A_{cam}\left[R \quad t\right]M \tag{1}$$

where A_{cam} indicates the intrinsic matrix of the camera, R and t are specified by the position of the viewpoint. In order to get the observed image m equal to the desired image $m_{desired}$, we need to prewarp the projector input image. The homography H describes the relationship between the projector input image (the desired image) and the viewpoint image via the projection surface as

$$m = Hm_{desired} \cdot \tag{2}$$

If we prewarp the projector input image as

$$m_{prewarped} = H^{-1}m_{desired} , \tag{3}$$

then, the projection image is undistorted in the viewpoint as

$$m = Hm_{prewarped} = H(H^{-1}m_{desired}) = (HH^{-1})m_{desired} = m_{desired} \cdot \tag{4}$$

4 Experimental Results and Discussion

A projector (SONY VPL-CX6) and a camera (PointGrey Dragonfly Express) were used in our experiments. They were synchronized to enable the camera to capture the image projected by the projector without frame loss. The images were at a resolution of 640 by 480 pixels.

In general, projection surfaces are not completely white so the augmentation may be modulated by the color of the surface. In our experiments, the projection surface was covered with color-textured sheets to emphasize the color distortion as shown in Fig. 3(d). However, the color distortion could be easily compensated using the photometric adaptation method [4]. With our framework, therefore, it was possible to project augmented reality images onto dynamic surfaces without both geometric and radiometric image distortion.

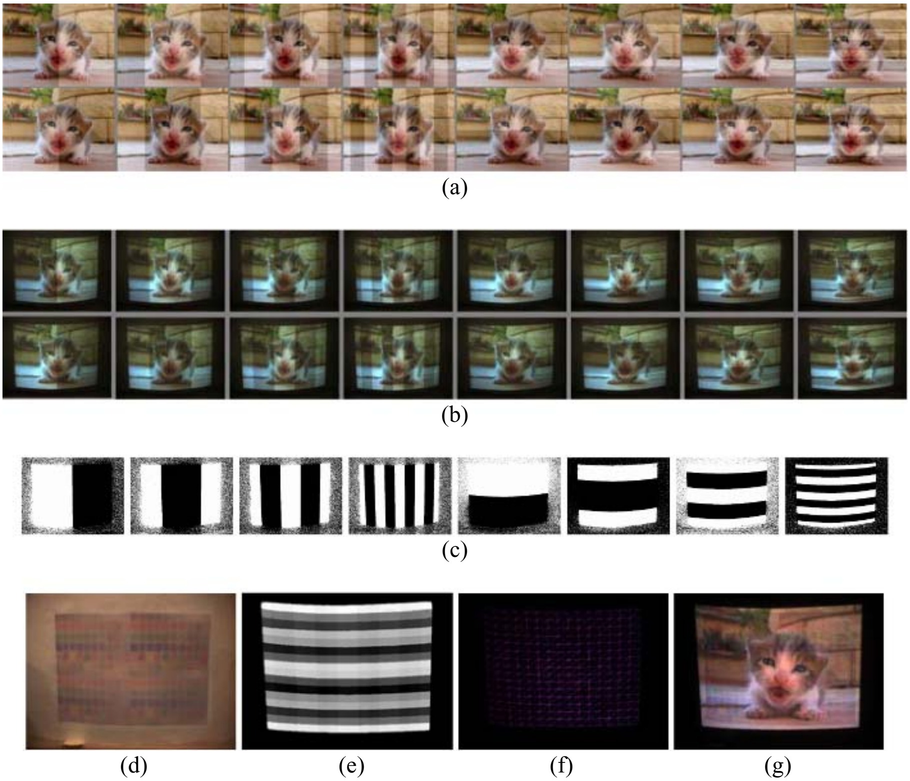
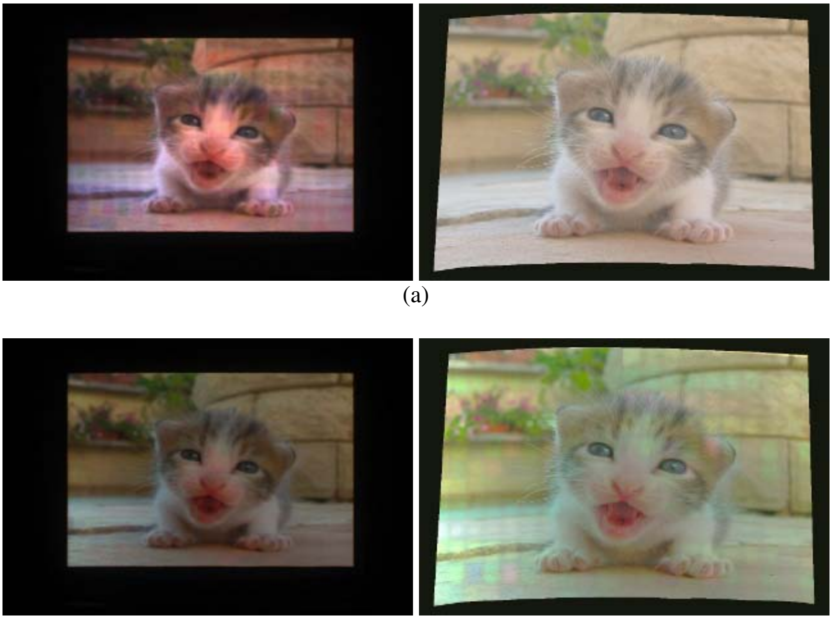


Fig. 3. Results of recovering the surface geometry using the binary-code method combined with the pattern embedding method. (a) projector input image in which the pattern images are embedded (top images: even frames, bottom images: odd frames), (b) images captured by a camera after projecting the projector images (top images: even frames, bottom images: odd frames), (c) the pattern images extracted from the camera images, (d) convex and color-textured screen used in our experiments, (e) image which visually represents the codes computed from the pattern images, (f) recovered geometry represented by triangular meshes, (g) the projection image which is distorted geometrically and radiometrically.

Figure 3 shows the process of recovering the surface geometry using the binary-code method combined with the pattern embedding method. One could not notice the existence of the pattern images when the even and odd frames in Fig. 3(a) were projected alternately at 60 frames per second. As shown in Fig. 3(c), however, the pattern images could be extracted from the absolute difference between the even and odd frames of the camera images in Fig. 3(b). The codewords, which were obtained from the sequence of the pattern images, were visually represented in Fig. 3(e). The correspondences between the projector image and the camera image were computed based on the codewords. The triangular meshes in Fig. 3(f) were obtained by applying a linear triangulation method to the correspondences.

When an image was projected onto a nonplanar and color-textured surface or whenever the surface geometry was changed, the resulting projection image was distorted geometrically and radiometrically as shown in Fig. 3(g). After performing the process of the surface geometry recovery, the geometric correction method and the radiometric compensation method [4] were applied to the projector input image together. Figure 4 shows the results of compensating the geometric and radiometric image distortion of the projection. At a given viewpoint, both the geometric image distortion and radiometric image distortion were completely compensated in the projection as shown in Fig. 4(b).

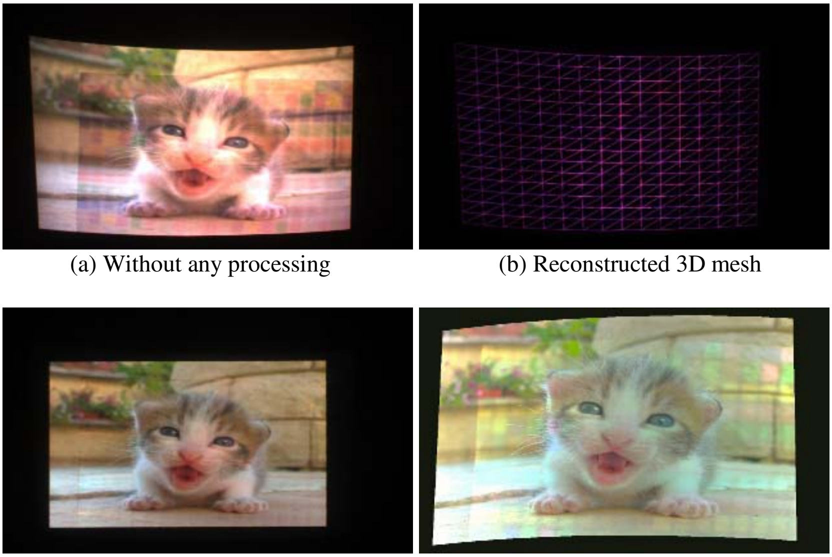


(a)

(b)

Fig. 4. Results of compensating the geometric and radiometric image distortion of the projection. (a) geometric compensation only, (b) radiometric compensation combined. The right images are the projector input images which are modified for geometric or radiometric compensation in advance.

Figure 5 and 6 show the results of adapting the projection to the dynamic change of the projection surface geometry. In the beginning, the geometric and radiometric image distortion of the projection in Fig. 5(a) was completely compensated in Fig. 5(c). When the surface geometry was changed, the projection was distorted again as in Fig. 6(a). The change could be recognized from the shape of the meshes in Fig. 5(b) and Fig. 6(b). However, the distortion was disappeared (compensated by the implicit process) immediately as shown in Fig. 6(c). The projector input image was continuously modified for adapting to the dynamic change of the projection surface geometry as shown in Fig. 5(d) and Fig. 6(d).



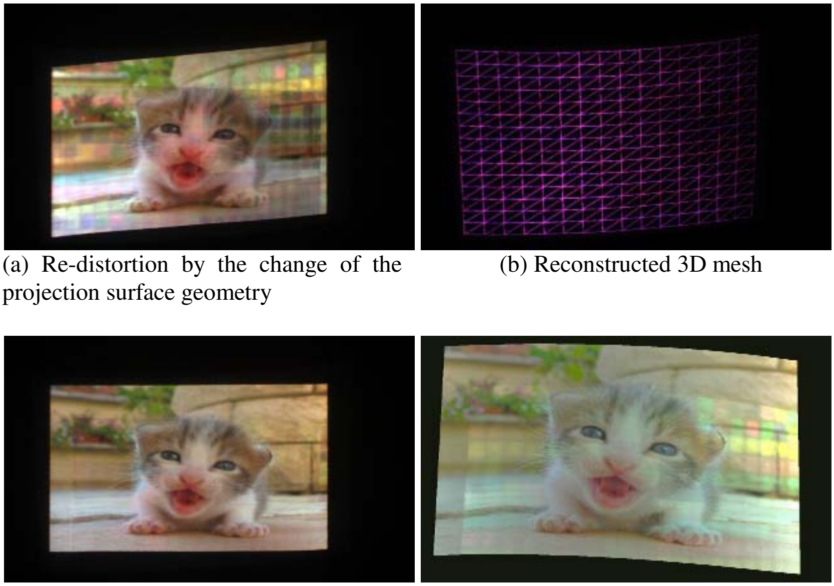
(a) Without any processing

(b) Reconstructed 3D mesh

(c) With geometric correction and radiometric compensation

(d) Modified projector input image

Fig. 5. Initial compensation of the geometric and radiometric image distortion of the projection



(a) Re-distortion by the change of the projection surface geometry

(b) Reconstructed 3D mesh

(c) With geometric correction and radiometric compensation

(d) Modified projector input image

Fig. 6. Adaptation to the change of the projection surface geometry. The projection is distorted by the change of the surface geometry for a while, but the projection is adapted immediately.

Comparing with Fig. 5(c) and Fig. 6(c), the projection looks like unchanged and one could not notice the change of the projection surface.

5 Conclusion

In this paper, we provided a simple and robust method for projecting augmented reality image without geometric image distortion onto dynamic surfaces. For this purpose, a new technique for embedding pattern image into the augmented reality image and thus making the pattern image invisible was proposed and its effectiveness was shown through the experimental results.

For projector-based augmented reality systems, radiometric compensation for dynamic surfaces is another crucial part. Fujii's method is available only when the optical characteristics of the projector and camera are not changed [4]. Of course, real-time radiometric compensation of the projection would be possible by embedding color pattern images into AR images even when the optical characteristics of the camera are changed. However, the simultaneous geometric and radiometric compensation requires too many pattern images to be embedded. Currently, we are trying to reduce the number of pattern images.

Acknowledgement. This study was supported by a grant(02-PJ3-PG6-EV04-0003) of Ministry of Health and Welfare, Republic of Korea.

References

1. Ashdown, M., Sukthankar, R.: Robust Calibration of Camera-Projector System for Multi-Planar Displays. Technical Report HPL-2003-24, HP Labs (2003)
2. Bimber, O., Raskar, R.: Spatial Augmented Reality. A K Peters (2005)
3. Cotting, D., Naef, M., Gross, M., Fuchs, H.: Embedding Imperceptible Patterns into Projected Images for Simultaneous Acquisition and Display. Proc. of ISMAR'04 (2004)
4. Fujii, K., Grossberg, M.D., Nayar, S.K.: A Projector-Camera System with Real-Time Photometric Adaptation for Dynamic Environments. Proc. of CVPR'05 (2005) 814-821
5. Hartley, R., Zisserman, A.: Multiple View Geometry. Cambridge University Press (2003)
6. Palmer, S.E.: Vision Science. The MIT Press (1999)
7. Park, H., Lee, M.-H., Kim, S.-J., Park, J.-I.: Surface-Independent Direct-Projected Augmented Reality. Proc. of ACCV'06 (2006) 892-901
8. Pocket Imager. http://www.samsung.com/uk/products/projectors/mobileprojector/spp300memxedc.asp
9. Projector-related papers. http://www.cs.unc.edu/~raskar/Projector/projbib.html
10. Raskar, R., Beardsley, P.: A Self Correcting Projector. Technical Report TR-2000-46, Mitsubishi Electric Research Laboratories (2002)
11. Salvi, J., Pages, J., Batlle, J.: Pattern codification strategies in structured light systems. Pattern Recognition, vol.37, no.4 (2004) 827-849
12. Sukthankar, R., Stockton, R., Mullin, M.: Smarter Presentations: Exploiting Homography in Camera-Projector Systems. Proc. of ICCV'01 (2001)
13. Vieira, M.B., Velho, L., Sa, A., Carvalho, P.C.: A Camera-Projector System for Real-Time 3D Video. Proc. of CVPR'05 (2005)
14. Yasumuro, Y., Imura, M., Manabe, Y., Oshiro, O., Chihara, K.: Projection-based Augmented Reality with Automated Shape Scanning. Proc. of SPIE EI'05 (2005)

Robust Moving Object Detection on Moving Platforms

Ming-Yu Shih and Bwo-Chau Fu

Advanced Technology Center,
Information & Computer Laboratories,
Industrial Technology Research Institute,
E000, Bldg.51,195 Sec.4, ChungHsing Rd, Chutung, Hsinchiu, Taiwan 310, R.O.C.

Abstract. Most moving object detection methods rely on approaches similar to background subtraction or frame differences that require camera to be fixed at a certain position. However, on mobile robots, a background model can not be maintained because of the camera motion introduced by the robot motion. To overcome such obstacle, some researchers proposed methods that use optical flow and stereo vision to detect moving objects on moving platforms. These methods work under a assumption that the areas belong to the interesting foreground moving objects are relatively small compare to the areas belong to the uninteresting background scene. However, in many situations, the moving objects may approach closely to the robot on which the camera is located. In such a case, the assumption of *small foreground moving object* will be violated. This paper presents a framework which shows that the small foreground moving object assumption could be relaxed. Further, it integrates the observations in motion field and image alignment to provide a robust moving object detection solution in unconstrained indoor environment.

1 Introduction

The conventional video surveillance cameras are cheap and easy to distribute, however, their effectiveness are restricted by limited human resources to observe the possible irregular or suspicious events. The governments and security agencies are increasingly exploring the potential of "intelligent video surveillance" to compensate the inadequacy of human resources. As a result, The goal to build smart video surveillance systems has become one of the main research topics for many academic and business agencies particularly of those that focus their research on computer vision related issues.

The Video Surveillance and Monitoring(VSAM) [1][2] system founded by Carnegie Mellon University allowed a single operator to monitor a cluttered environment using multiple cooperative cameras. Pfinder [3] of MIT used a multi-class statistical model of color and shape to obtain a 2D representation of human hand and head to track people and interpret the tracked people's behaviors in a wide range of viewing conditions. Video Surveillance cooperations such

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 591–600, 2006.

as *ObjectVideo* and *iOmniScient* have extended further the video surveillance techniques to commercialized autonomous surveillance systems that monitored diversified areas such as airport, railways, museums, landmarks, and borders in applications from intruder detections to abandoned object detection.

Recently, the trend of video surveillance systems has shifted from conventional fixed camera frameworks to more flexible, distributed mobile platforms, for example, self-guided security robots. To estimate the 3D camera motion(the ego-motion), Irani [4][5] computed the dominant 2D parametric motion between two frames to register the images to remove all effects of camera rotation and then used the epipolar field and the computed 3D translation along with the detected 2D parametric motion to recover the 3D camera motion. Lowe [6] used scale-invariant feature transform(SIFT) that are invariant to image translation, scaling, and rotation to be landmarks suitable for mobile robot localization and map building applications. Talukder [7][8] established a framework of moving object detection from moving vehicles using dense stereo and optical flow in a dynamic scene. However, conventional video surveillance topics such as moving object detection, tracking, human motion analysis, or activity recognition have became even tougher on the moving platform due to the special nature of moving camera circumstances.

In this paper, our framework relaxed the small foreground assumption to allow the moving object to move closely to the robot. This framework utilizes stereo and optical flow information to detect mismatches between the predicted and observed optical flow [9] and integrates optical flow and image alignment observations to provide robust moving object detections on moving platforms.

In section 2, we explain the ego-motion estimation process that is reliable under dynamic indoor environment. In section 3, the optical flow mismatch detection and its integration with image alignment to detect moving object is presented. The experimental results on odometry estimation and moving object detection are shown in section 4. Finally, we conclude the result and discuss the future works in section 5.

2 Ego-Motion Estimation

Under the assumption that there is only one, rigid, relative motion between the camera and the observed scene, and the illumination conditions do not change abruptly, the camera ego-motion could be realized by establishing the relation between the 2D optical flow $[u\ v]$, the 3D object point P, and the relative 3D motion between P and the camera center C.

2.1 Velocity Field Projected Model

The 3D displacement of the observed scene as well as the 3D displacement of a rigid object in the Cartesian coordinates can be modeled by camera rotation and translation of the form

$$P' = RP + T \tag{1}$$

where R is a 3×3 rotation matrix, $T = [T_x\ T_y\ T_z]^T$ is a 3D translation vector, and $P = [X\ Y\ Z]^T$ and $P' = [X'\ Y'\ Z']^T$ denotes the coordinates of an object point at time t and t' in the 3D scene with respect to the center of rotation. By expressing the rotation matrix in terms of the Eulerian angles, θ, ψ, and ϕ that denote arbitrary rotations in the 3D space about the X, Y, and Z axes respectively, we can form the displacement model in the form

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & -\varDelta\phi & \varDelta\psi \\ \varDelta\phi & 1 & -\varDelta\theta \\ \varDelta\psi & \varDelta\theta & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{2}$$

By taking the time derivative of the 3D displacement model, the $3D$ velocity model can be obtained to represent the velocity of a point X in the 3D space as

$$\begin{bmatrix} \widetilde{V_x} \\ \widetilde{V_y} \\ \widetilde{V_z} \end{bmatrix} = \begin{bmatrix} 0 & -\varOmega_z & \varOmega_y \\ \varOmega_z & 0 & -\varOmega_x \\ -\varOmega_y & \varOmega_x & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{3}$$

where $\varOmega = [\varOmega_x\ \varOmega_y\ \varOmega_z]^T$ is the angular velocity vector and $V = [V_x\ V_y\ V_z]^T$ is the translational velocity vector. For perspective cameras with focal length f, the 2-D image plane projection, $p = [x\ y]^T$, of the 3D object point in space, $P = [X\ Y\ Z]^T$, is

$$x = \frac{fX}{Z},\ y = \frac{fY}{Z} \tag{4}$$

Equation (4) is time derived on both sides to obtain the relation between the velocity of P in space and the corresponding velocity of p, 2D optical flow $[u\ v]$, on the image plane,

$$u = \frac{1}{Z}(f\frac{dX}{dt} - x\frac{dZ}{dt}),\ v = \frac{1}{Z}(f\frac{dY}{dt} - y\frac{dZ}{dt}). \tag{5}$$

Further, using equation (3) and (5), the relation between the 2D optical flow $[u\ v]$, the 3D object point P, and the relative 3D motion between P and the camera can be obtained

$$\begin{aligned} u &= \frac{1}{Z}(f\widetilde{V_x} - x\widetilde{V_z}) \\ &= \frac{f}{Z}(\varOmega_y Z - \varOmega z Y + V_x) - \frac{x}{Z}(\varOmega x Y - \varOmega y X + V_z) \\ &= \frac{f}{Z}V_x - \frac{x}{Z}V_z - \frac{xy}{f}\varOmega_x + \frac{(f^2 + x^2)}{f}\varOmega_y - y\varOmega_z. \\ v &= \frac{f}{Z}V_y - \frac{y}{Z}V_z - \frac{(f^2 + y^2)}{f}\varOmega_x + \frac{xy}{f}\varOmega_y - x\varOmega_x. \end{aligned} \tag{6}$$

or expressed in matrix form

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \frac{1}{f}\begin{bmatrix} -xy & (f^2 + y^2) & -fy \\ -(f^2 + y^2) & xy & fx \end{bmatrix} \begin{bmatrix} \varOmega_x \\ \varOmega_y \\ \varOmega_z \end{bmatrix} \tag{7}$$

2.2 Robust Model Parameter Estimation

The assumption that there is only one, rigid, relative motion between the camera and the observed scene can not be hold with the presence of moving objects. The moving pixels, *outliers*, that are introduced into the image plane by the foreground moving objects are useless in the computation of the 6-Degree of freedom ego-motion parameters. Most approaches handle such exception by assuming the number of moving foreground pixels *outliers* are relatively fewer than that of the static background pixels *inliers*. These methods therefore use as much of the data as possible to obtain an initial solution and then eliminate the invalid data points iteratively by minimum least square distance function. However, in surveillance applications such as indoor surveillance robot, the moving object could approach closely to the robot on which the camera is located and violate the assumption.

To eliminate the erroneous effects introduced by the *outliers* and to determine a set of useful *inliers*, a robust estimation algorithm is essential to apply. There are many robust algorithms available but they are different on their abilities to resist large proportion of outliers. In our case, the presence of closely moving objects introduce large number of outliers. Conventional methods such as least mean-square error (LMSE) are prone to error especially at the presence of outliers. Therefore, algorithms such as RANdom SAmple Consensus (RANSAC), which use as small an initial data set as possible and enlarges this set only when better estimation can be obtained, are better applied in our case.

First, RANSAC randomly select a sample of s minimal data points from the complete data set S and use the sample set to instantiate the initial motion parameters. With these initial motion parameters, the number of data points S_i that are fitted with the motion parameters within a distance threshold t is recorded. Then, after repeated this process N times, the largest consensus set S_i that has the largest number of within distance data points is selected and the corresponding motion parameters is determined as the ego-motion parameters of the moving platform.

In our implementation, the number of trials N, is computed so that

$$N = \log(1-p)/\log(1-(1-\epsilon)^s). \tag{8}$$

where s is set to 3, the minimum number of sample points needs to estimate the motion parameters. We choose p to be 0.99 to make sure a probability of 99% that at least one of the s random sample data points is an inlier can be achieved.

3 Moving Object Detection on the Move

Optical flow computation involves estimates of the motion field under the the data conservation and spatial coherence constraints. However, in unconstrained real environment, the brightness constancy and spatial smoothness assumptions are frequently violated because of the presence of multiple motions in the observed image sequences. To address such violations, the framework proposed by

Black [9] that has been proven to work well in such cases is chosen to be utilized in our implementation.

The moving object detection on the move involves the extraction of moving objects from the 2D optical field which corresponds to the projection of the 3D velocity filed on the 2D image plane where the 2D motions are generated by the imaged dynamic object movement in the scene or by the motion of the robot platform on which the camera is placed. To fix the 2D motion generated by the camera ego-motion and identify the optical flow discontinuously caused purely by the object movement in the scene, the predicted optical is estimated using motion parameters obtained from the ego-motion estimation to detect the motion field discontinuities between the predicted and observed optical flow, estimated from the observation of consecutive frames.

Detect moving object with the discontinues on motion filed alone would sometimes generate incomplete detection result because of similar camera and object motion that produce non-obvious responses on motion filed discrepancies. Therefore, the information of discontinuities detected in motion field and image alignment are integrated to enhance a more complete detection result.

3.1 The Discontinuities in Motion Field

To fix the 2D motion generated by the camera ego-motion, the predicted optical flow could be computed using equation (7) along with the camera intrinsic parameters. For a given reference image frame $I(t)$ at time t, the discontinuities in the motion field that represent the projected 2D motion which introduced purely by the dynamic object movement in the 3D scene are then given by the differences between the predicted and observed optical flows

$$\begin{aligned} u_i &= u_i^e - u_i^o, \\ v_i &= v_i^e - v_i^o. \end{aligned} \tag{9}$$

where $i = 1 \ldots N$, the total number of pixels of image $I(t)$, $[u_i^e \; v_i^e]$ is the predicted 2D optical flow of pixel p_i along its x and y directions, respectively, $[u_i^o \; v_i^o]$ is the observed 2D optical flow of the same pixel p_i, and $[u_i \; v_i]$ is the resulting discontinuity magnitude between the predicted and observed optical flow.

3.2 The Discontinuities in Image Alignment

The discontinuities in image alignment is computed to integrate with the mismatches between the predicted and observed optical flow to promise highly accurate moving object detections in the observing image sequences. For each consecutive image frame pairs $I(t)$ and $I(t-1)$, the discontinuities $f_i(t)$ at time t of each pixel p_i at coordinate (x, y) in image alignment could be obtained by

$$f_i(t) = I(x + u_i^e, y + v_i^e, t + 1) - I(x, y, t). \tag{10}$$

where $I(x, y, t)$ is the intensity value of pixel p_i at coordinate (x, y) in the image plane at time t.

3.3 Discontinuity Integration Using Gaussian Modeling

The information of discontinuities detected in motion field and image alignment are integrated to allow highly accurate detection of moving object on the move. The optical flow magnitude d_i of pixel p_i is quantized into L empirical decided levels to q_i by

$$q_i = \lfloor \frac{d_i L}{D} \rfloor \tag{11}$$

where $d_i = \sqrt{u_i^2 + v_i^2}$, and D is the maximum magnitude of d_i, where $i = 1 \ldots N$. In our case, we quantized the magnitude to 11 levels. Using the quantized optical flow, the image I is segmented into K regions

$$I = \bigcup_{k=1}^{K} R_k \tag{12}$$

where $R_k = \{p_1^k, p_2^k, \ldots, p_n^k\}$ with constraints that all pixels p_i^k in R_k are four connected and $q_1^k = q_2^k = \ldots = q_n^k$. Next, a gaussian model is used to model the histogram of f_i of image I to obtain the standard deviation σ and mean m of f_i. A region R_k is identified as moving object if

$$\exists f_i^k - m > 1.96\sigma \text{ and } q_i^k > t \tag{13}$$

where t is an empirical decided threshold value.

4 Experimental Result

In our experiments, the Digiclops stereo vision device is used to obtain scene structure of the observed scene. The Digiclops establishes correspondences between images using the sum of absolute differences correlation method with a correlation mask size which is set to 11×11 in our case. To ensure a highly accurate scene structure estimation, Digiclop sub-pixel interpolated surface, texture and uniqueness validation constraints are applied in our experiments to filter out uncertain scene structure estimations. In the parameter settings, the texture validation determines whether disparity values are valid based on levels of textures in correlation mask; the uniqueness validation determines whether the best match for a particular pixel is significantly better than other matches within the correlation mask; the surface validation validate regions of a disparity image based on an assumption that they must belong to a likely physical surface in the image.

4.1 Ego-Motion Estimation

Both LMSE and RANSAC are implemented with Black and Kanade-Lucas-Tomasi(KLT) optical flow trackers to compare their performance. It can be seen from Fig.1(a)(b)(d)(e), LMSE tend to mistakenly mark foreground moving pixels as *inliers* and use them in the Ego-Motion Estimation. In contrast, RANSAC is resistance to these foreground motions and use only background pixels to estimate the Ego-Motion.

In addition, we have observed that, in some cases as those shown by Fig.1(c)(f), Black's optical flow algorithm outperforms the KLT method. This could due to the locality constraint imposed by Black's optical flow algorithm and this constraint helped Black's optical flow to resist lighting variation between frames. Therefore, the combined use of Black's optical flow and RANSAC are applied in all of our experiments.

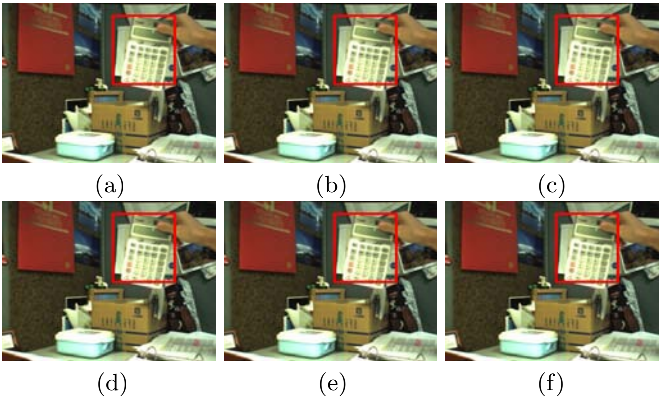


Fig. 1. The ego-motion estimations using different feature matching and estimation pairs are compared (a) KLT + LMSE (b) KLT + RANSAC (c) Detection result using KLT + RANSAC (d) Optical Flow + LMSE (e) Optical Flow + RANSAC. (f) Detection result using Optical Flow + RANSAC. The green areas are the *inliers* and the blue areas are *outliers*. Only inliers are used in the ego-motion computation.

4.2 Integrated Moving Object Detection

The discontinuities between the observed and estimated optical flow could be used to locate the foreground moving object which does not belong to the static background scene. As shown in Fig.2, with the presence of object motion and camera motion, the area of moving objects could be clearly seen from the responses of mismatches between the observed and estimated flow. Motions with different intensities and directions are depicted in black vectors that are plotted in five pixels intervals. The red pixels are corresponding to invalid scene structure computation due to Digiclop validation constraints.

The information of discontinuities detected in motion field and image alignment are integrated to enhance a more complete detection result as shown at Fig.3. However, the hand motion are filtered out in the scene structure estimation preprocess because of the unstable stereo matching on hand area.

In Fig.4, a robust and complete detection result could be generated even with the presence of significant object motion that occupied most areas of the

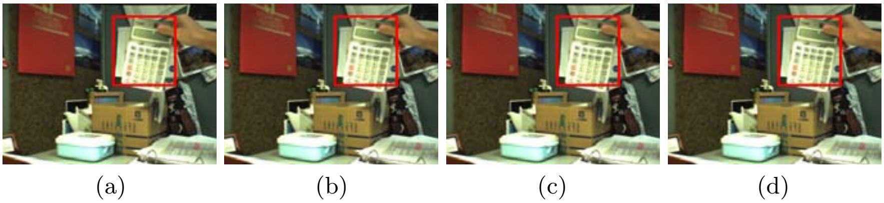

(a) (b) (c) (d)

Fig. 2. (a) the original image (b) the observed optical flow using Black's method(c) the estimated optical flow computed using ego-motion parameters (d) the discontinuities between the observed and estimated optical flow

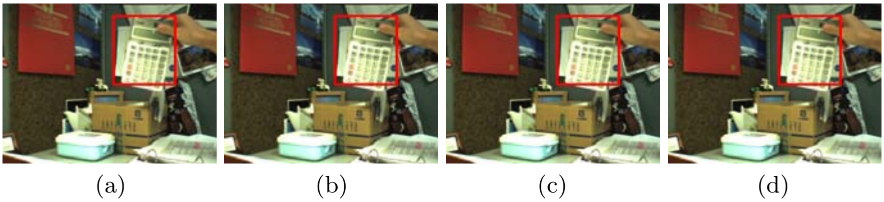
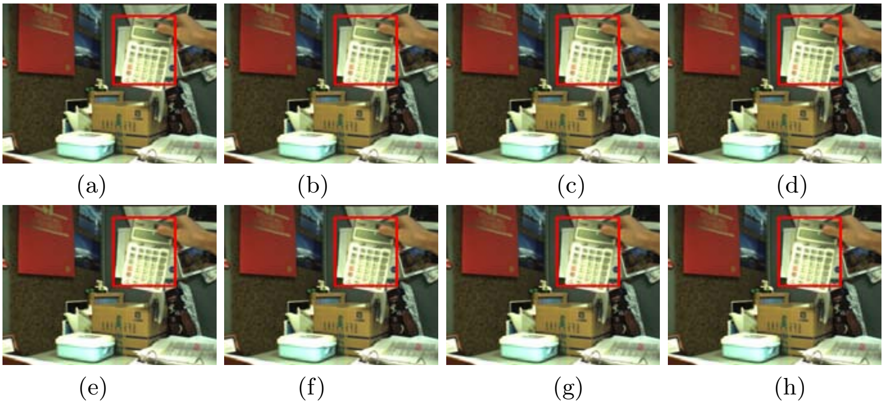

(a) (b) (c) (d)

Fig. 3. (a) the discontinuities between the observed and estimated flow (b) the detected result on motion filed discontinuities (c) the discontinuities on image alignment (d) the detected result on integrated discontinuities

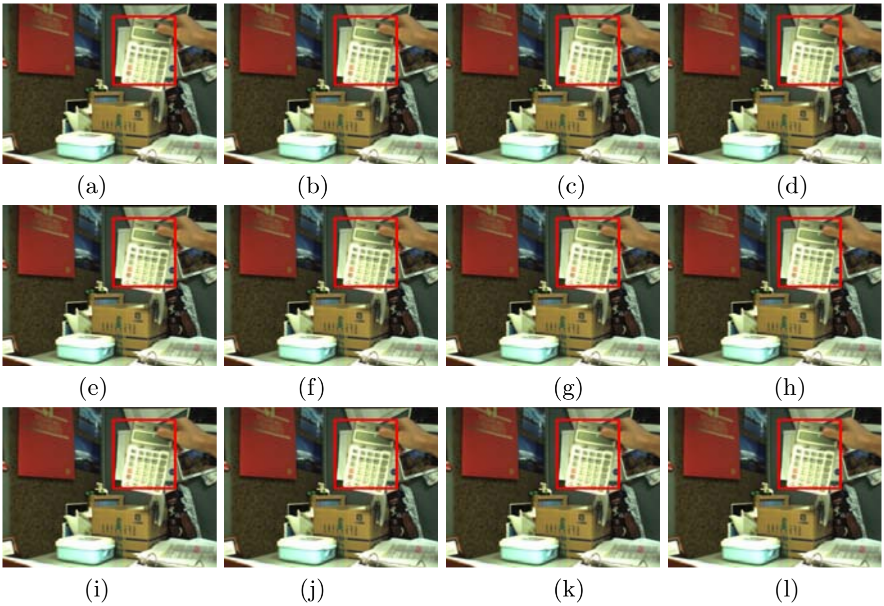

(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4. From (a) to (h), a moving object approaching the camera at the presence of camera motion

image. In most other proposed algorithms, this usually generate false ego-motion estimation and therefore produce incorrect object detection result.

The experiment is further deployed on the Pioneer-DX3 mobile platform to detect moving object on the move in unconstrained environment as shown in Fig.5.



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

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 5. In these video sequences, the camera installed robot was moving toward the walking person. From (a) to (d), the forward moving robot detected a person walking across the scene; from (e) to (h), the robot accurately detected the body of the person and the car that he was pushing; from (i) to (l), the robot can still provide accurate detection result when the person was walking toward it.

5 Conclusion and Future Works

In this paper, we integrated the stereo and optical flow information to provide moving object detection ability for moving robot platforms. The inspiring results encouraged us to extend our current research to applications such as suspicious human movement detection, people pursing, and gesture commanding on the move. However, because of the time consuming optical flow estimation, our current framework can not be executed on real time. To make the framework practical, a real time robust optical flow estimation will be needed. Further, an efficient and reliable stereo matching algorithm should also be developed to relax current validation constraints imposed by the Digiclop stereo vision device.

Acknowledgment

This work is a partial result of Project 5301XSY320 conducted by Industrial Technology Research Institute under sponsorship of the Ministry of Economic Affairs, R.O.C, Taiwan.

References

1. R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa , A System for Video Surveillance and Monitoring, tech. report CMU-RI-TR-00-12, Robotics Institute, CMU, May, 2000.
2. X. Zhou, R. Collins, T. Kanade, and P. Metes," A Master-Slave System to Acquire Biometric Imagery of Humans at Distance," ACM International Workshop on Video Surveillance. November (2003)
3. Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. IEEE Trans. Pattern Analysis and Machine Intelligence, 19(7):780–785, July 1997.
4. L. Zelnik-Manor and M. Irani, Multi-Frame Estimation of Planar Motion . IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Vol. 22, No. 10, pp. 1105-1116, October 2000.
5. M. Irani, B. Rousso, S. Peleg, Recovery of Ego-Motion Using Region Alignment . IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Vol. 19, No. 3, pp. 268–272, March 1997.
6. S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. Intl. Journal of Robotics Research, 21(8), 2002.
7. A. Talukder, L. Matthies,"Real-time detection of moving objects from moving vehicles using dense stereo and optical flow," IEEE Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, September (2004).
8. A. Talukder, S. Goldberg, L. Matthies, A. Ansar, "Real-time detection of moving objects in a dynamic scene from moving robotic vehicles," IEEE Conference on Intelligent Robots and Systems, Las Vegas, NV, October 2003.
9. Black M, Anandan P. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. Computer Vision and Image Understanding 1996.

Automatic Selection and Detection of Visual Landmarks Using Multiple Segmentations

Daniel Langdon, Alvaro Soto, and Domingo Mery

Pontificia Universidad Catolica de Chile
Santiago 22, Chile
dlangdon@puc.cl, {asoto, dmery}@ing.puc.cl

Abstract. Detection of visual landmarks is an important problem in the development of automated, vision-based agents working on unstructured environments. In this paper, we present an unsupervised approach to select and to detect landmarks in images coming from a video stream. Our approach integrates three main visual mechanisms: attention, area segmentation, and landmark characterization. In particular, we demonstrate that an incorrect segmentation of a landmark produces severe problems in the next steps of the analysis, and that by using multiple segmentation algorithms we can greatly increase the robustness of the system. We test our approach with encouraging results in two image sets taken in real world scenarios. We obtained a significant 52% increase in recognition when using the multiple segmentation approach with respect to using single segmentation algorithms.

1 Introduction

Vision is an attractive option to provide an intelligent agent with the type of perceptual capabilities that it needs to deal with the complexity of an unstructured natural environment. The robustness and flexibility exhibited by most seeing beings is a clear proof of the advantages of an appropriate visual system. In particular, the selection and detection of relevant visual landmarks is a highly valuable perceptual capability. In effect, the ability to select relevant visual patches from an input image, such that, they can be detected in subsequent images, is a useful tool for a wide variety of applications, such as video editing, place and object recognition, or mapping and localization by a mobile agent.

In this paper we present an unsupervised method for the automatic selection and subsequent detection of suitable visual landmarks from images coming from a video stream. To achieve this goal, we frame the problem of landmark detection as a pure bottom-up process based on low level visual features such as shape, color, or spatial continuity. The goal is to select interesting, meaningful, and useful landmarks.

We base our approach on the integration of three main mechanisms: visual attention, area segmentation, and landmark characterization. Visual attention provides the selection mechanism to focus the processing on the most salient parts of the input image. This eliminates the detection of irrelevant landmarks

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 601–610, 2006.

and reduces computational costs significantly. Area segmentation provides the detection mechanism to delimite the spanning area of each salient region. This spanning area defines the scope of each relevant landmark. Finally, the landmark characterization mechanism provides a set of specific features for each landmark. These features allow the system to recognize and distinguish each landmark from others in subsequent images.

Considerable effort has been put on finding solutions to each of the individual visual mechanisms mentioned above. Interesting results have been found [13] [11], but the interaction of these steps has not been deeply pursued. In our study, we compare the set of landmarks detected in a image with a database of previously stored landmarks. In this case, a poor integration of attention, segmentation, and characterization reduces the efficiency and robustness of the algorithm.

In particular, we believe that an incorrect segmentation of a landmark produces severe problems in the next steps of the image analysis. An inaccurate segmentation may cause that the characterization of the landmark is imprecise, more complex, or even erroneous. For example, an incorrect segmentation of a landmark, that includes parts of the background in the segmented area, may lead to situations where the system only detects the landmark when the specific background is present, or even worse, can lead to situations where the system confuses the landmark with parts of the background image.

This paper addresses the integration problem, focusing on the segmentation problem mentioned above. Our hypothesis is that no segmentation algorithm can work correctly in every situation, so multiple segmentation algorithms need to be used to obtain a correct landmark from the attended spot. The best segmentation can then be used in the future to avoid the mentioned problems.

This document is organized as follows. Section 2 provides background information and describes previous work on the three processing steps mentioned. Section 3 presents the proposed method and relevant implementation details. Section 4 presents our empirical results. Finally, Section 5 presents the main conclusions of this work and future work on this topic.

2 Previous Work

In the computer vision literature, there is an extensive list of works that individually target the problems of attention, segmentation, and characterization. We briefly review some relevant works in each area. We also review some relevant works in the area of landmark selection and detection.

Attention is the process of selecting visual information from an image based on a measurement of saliency. Previous work in this area includes several models of visual attention. Tsotsos et al. [12] selectively tuned neuron models at the salient location with top-down mechanisms, and winner-take-all networks. Itti et al. [4] introduced a model for selecting locations from a saliency map according to decreasing saliency. Sun and Fisher [11] extended Duncan's Integrated Competition Hypothesis [2] with a framework for location-based and object-based attention using grouping.

Segmentation is the process of identifying regions of an image that share some common characteristics like color, texture, shape, etc. There are many techniques used in segmentation and a lot of research is being done in the topic every day. Common approaches to the problem include thresholding, clustering, region growing, and boundary-based techniques. Given space constraints, we refer the reader to [8] for a comprehensive overview and comparison of segmentation methods.

Characterization is the process of finding a set of descriptors or features that provide an easy and robust identification of each landmark. Several algorithms have been used for this purpose; structural descriptions [1], texture and color descriptors (such as histograms) [9], among others. One of the most relevant trends in this area tries to find the presence of stable features in the input image. These features should be stable even with slight variations on the input, such as, changes in lighting conditions, field of view, or partial occlusions. This approach became very popular after Harris and Stephens [3] presented their corner detector. Recently, Lowe [6] presented a refinement of this idea, called the Scale Invariant Feature Transform (SIFT), which achieved a lot of popularity in recent years due to its success in several applications.

In the context of landmark selection and detection, most of the previous approaches focus mainly on the detection problem [10]. In these cases, the selection process operates in a supervised way using visual models for specific, manually selected, landmarks. Recently, Karlsson et al. [5] presented a solution for the robot localization and navigation problem based on SIFT features. This system is capable of creating visual landmarks dynamically and of recognizing these landmarks, afterwards, with enough robustness to provide real-time robot localization. In contrast to our approach, this work does not use attention mechanisms and a landmark is characterized by the SIFT features of the complete image. Walther et al. [14] used a similar approach to detect moving objects with a remotely operated underwater vehicle equipped with a video camera. The influence of saliency on recognition was also proven by Walther et al. [13], but the influence of the segmentation process has not been determined yet.

3 Our Approach

As mentioned earlier, our goal is to create an unsupervised method for the selection and subsequent detection of visual landmarks. We believe that a correct segmentation can help to avoid false detections and wrong data associations. Our hypothesis is that no single segmentation algorithm may provide enough robustness to achieve satisfactory results.

In this section, we present a method that integrates landmark selection, via attention guided search for salient image locations, and landmark detection, via multiple segmentations around the relevant image locations. As an additional step, each segmented landmark is characterized by a set of highly discriminative visual features that facilitates detection and avoids problems related to wrong data associations. We explain in detail each of the main steps of our approach.

3.1 Attention

We use the bottom-up saliency map of Itti et al. [4] to extract salient locations from an input image. In our implementation, we modify the original algorithm by introducing an adaptive scheme that dynamically selects an appropriate number of relevant salient locations. We select this algorithm because of its emulation of primate attention mechanisms, which provides landmarks that are consistent with the kind of objects a human being would find important. This feature facilitates the evaluation of the relevancy of the detected landmarks. A brief explanation of the algorithm follows.

Each image is processed to extract multi-scale maps of orientation, intensity, and color. After computation of center-surround differences, the algorithm creates feature maps for each scale. These maps are then combined to form a color map, an intensity map, and an orientation map. The maps are grayscale representations, where bright areas mean highly salient locations. A single saliency map is obtained by merging these individual conspicuity maps. Once the saliency map has been extracted a winner-take-all (WTA) neural network is used to extract the image coordinates of the center of the most salient location. Then an inhibition-of-return method is applied to prevent the selection of the same location multiple times. The WTA neural network iterates several times over the saliency map obtaining the potential location of the relevant landmarks.

In the original implementation, the algorithm has no limitations on the number of locations to obtain. The saliency map is normalized after each iteration, therefore, a new salient location can be found every time. Previous approaches have fixed the number of locations to find in each image [13]. The problem is that images from different scenarios produce a highly variable number of relevant locations, so an adaptive scheme is needed.

In our implementation, we adapt the number of locations detected by limiting the evolution of the WTA neural network that finds each salient location. The key observation is that the evolution time required by the network to find a salient location is proportional to the intensity and distribution of saliency in the saliency map. In other words, the network will take less time to evolve on an image with distinctive and interesting objects. In this way, we calibrate the evolution time on the WTA network according to the distribution of bright pixels in the saliency map using training images. As a training criteria we use the average evolution time to match the number of relevant landmarks detected by a human operator in the training images.

3.2 Segmentation

Although a large amount of research has been made on segmentation, there is not yet a complete solution to cope with landmark segmentation in unstructured environments. Every segmentation algorithm can cope with certain situations but fail to produce an adequate result in others. Since we can not predict the situations the input can produce a priori, we can use multiple segmentation algorithms to increase the adaptability and robustness of our landmark detection algorithm.

We select two existing segmentation algorithms to find the area defined by the underlying landmarks. Our goal is to prove that the quality of the segmentation greatly influences the posterior usefulness of each landmark and that, depending on the situation, one segmentation criteria can produce a better result than the other. Both algorithms were selected because of their simplicity and good computational performance. Each of them relies on highly independent visual information, therefore, we expect that their behaviors may be different depending on input conditions. We refer to these two algorithms as the color based segmentation algorithm and the saliency based segmentation algorithm.

The color based segmentation algorithm is based on a technique proposed in [7] to segment color food images. The original algorithm includes three main steps. First, a grayscale image is obtained from the input using an optimal linear combination of the RGB components found in the pixels around a specific image location. Then, a global threshold is estimated using a statistical approach. Finally, a morphological operation is used to fill possible holes that may appear in the final segmented area.

In terms of our application, one of the advantages of the previous algorithm is that it does not segment the full input image but only extracts a single object from it. Furthermore, one of the main problems of the original algorithm is finding a suitable image region to calibrate its color model. This is easily solved in our implementation by automatically calibrating the parameters of the color model using as foreground the area around a salient location.

The saliency based segmentation algorithm is based on a technique proposed in [13]. This algorithm was designed to work directly with the attention model used in our approach, so its application to our case is straightforward. In its operation, the algorithm finds the feature map that has the greatest influence over the selected part of the saliency map. Since in our case, the feature maps were already computed, this is done very efficiently. Then, the binary segmentation is extracted, re-scaled to match the saliency map, and smoothed. The resulting shape is a good approximation to the area occupied by the underlying landmark, or at least the part of the object that caught the attention in the first place. Although this algorithm includes a color saliency map in its process, the color model differs from the model used by the color based segmentation algorithm.

3.3 Characterization

For the characterization of the segmented patches, we use the SIFT feature extraction algorithm [6]. This algorithm provides highly discriminative features that, to some extent, are robust to the presence of affine distortion, noise, changes of viewpoint, and changes in illumination. Furthermore, when several features are extracted from a single object, the redundancy in information provides a robust detection even when only a subset of the features are visible due to partial occlusion. For details about the algorithm, we refer the reader to the original paper [6].

3.4 Integration

To accomplish our goal of unsupervised selection and subsequent detection of landmarks, we need to integrate the three steps described above. Figure 1 shows our complete integration scheme.

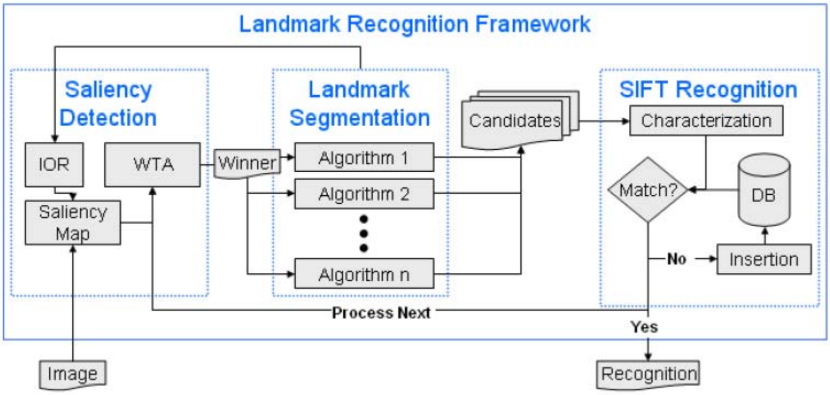


Fig. 1. Schematic operation of our approach. It integrates attention, segmentation, and characterization mechanisms for unsupervised recognition of relevant visual landmarks.

An input image is received and then its saliency map is computed by the attention algorithm. The first salient location is extracted, and the saliency and color segmentation algorithms are used to extract landmark candidates. Inhibition of return is calculated from the shape of the segmented landmarks and applied to the saliency map. This avoids selecting the same location in posterior iterations. The previous steps are repeated until the time to evolve the WTA network indicates that there are no more relevant salient regions to consider.

After the segmentation, SIFT features are extracted for each of the candidate landmarks. These features are then compared to the features of the landmarks in our database, which at the beginning of the process is empty. If there is a match, we modify our record about the number of times the landmark has been successfully recognized by the system. Otherwise we add both candidates to the database provided that certain constraints are satisfied. Our premise is to keep in the database only landmarks that are highly distintive and easy to detect. According to this, for each landmark we just accept SIFT features whose strength is about a fixed threshold. Furthermore, we require the number of relevant SIFT features found in each landmark to surpass a given threshold. In our algorithm, we require a minimum of 12 SIFT features to add a landmark to the database. Our experiments indicate that landmarks with fewer features usually do not produce enough matches to trigger a robust detection.

4 Results

4.1 Test

We created two different video sequences to test our algorithm. The first sequence corresponds to 185 images taken from three different rooms inside a house with a digital camera, at a resolution of 640x480 pixels (See Figure 2a). To test the robustness of the algorithm we consider images from different viewpoints, rotations, illuminations, and even blur produced by the motion of the camera. The second sequence consists of images taken in an indoor office environment (See Figure 2b). In this case, 875 images where automatically taken by a mobile robot when it navigated around a large section of corridors and a main hall. This set of images do not feature much illumination changes but they show moving objects.

4.2 Results

We independently applied our approach to each of the video sequences mentioned above. Figures 2a) and b) show images that highlight the typical landmarks detected in each sequence. Figure 2c) shows two landmarks that are aggregated to the database. Figure 2d) shows the detection of these landmarks in a posterior frame captured from a different position. Rectangles with identification labels are superimposed around new landmarks (ADD) and recognized landmarks (REC). After processing the full input, we obtained a database containing 258 landmarks for the home test set, divided between saliency landmarks and color landmarks. The office test set, meanwhile, produced a total of 535 landmarks. These databases were dynamically created in a complete unsupervised way by our system.

To test the relevance of using multiple segmentations, we counted the number of successful recognitions achieved by each segmentation algorithm over each landmark added to the database. Here, it is important to note that for a given salient region, each segmentation algorithm provides a different image patch to define the corresponding landmark.

Figure 3 shows the number of recognitions for some of the landmarks in the home data set. The differences in the heights of each pair of adjacent bars show that no segmentation algorithm is clearly the best option. We observed that in both training sets, recognition results vary considerably depending on the segmentation algorithm and the attended location.

Next, we compared the recognition results achieved by using both segmentation algorithms together with respect to using each segmentation algorithm alone. We considered the number of times that each location was recognized only by saliency landmarks, only by color landmarks, or by both of them. We counted the case in which both segmentations recognize the landmark as one detection. The results are shown in Table 1 and Table 2.

These results show that a multi-segmentation approach can add substantial robustness to the landmark recognition. Although, the cooperative integration

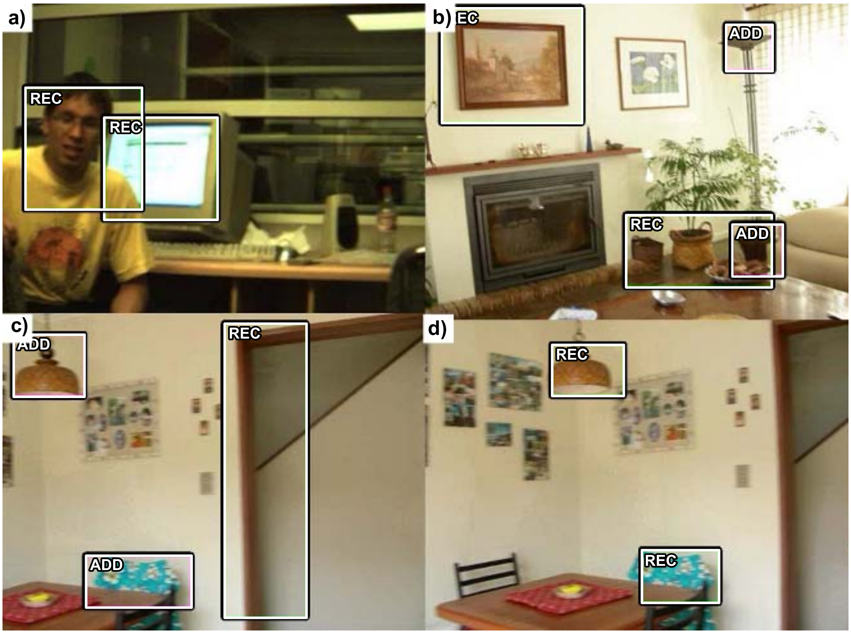


Fig. 2. a-b) Example images from both test sets. Relevant landmarks on each image are correctly detected. c) Two detected landmarks are added to the database. d) The two detected landmarks are recognized in a posterior frame taken from a different position.

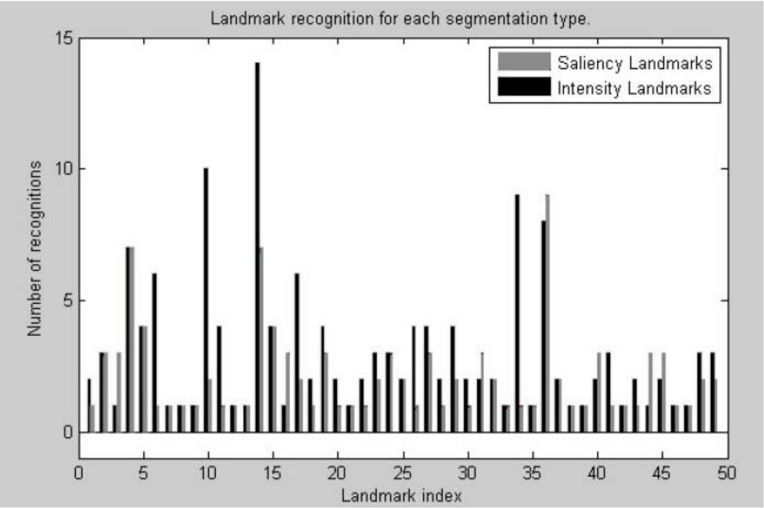


Fig. 3. Landmark recognitions for the home data set. Recognition depends on the segmentation algorithm used.

Table 1. Landmark recognitions

	Segmentation		
DataSet	Only Saliency	Only Color	Both
Home	248	270	358
Office	272	275	473

Table 2. Increase in recognition by using both segmentations

	Increase		
DataSet	Over Saliency	Over Color	Average
Home	44.35	32.59	38.47
Office	73.90	60.34	67.12
Average	59.12	46.46	52.79

of the segmentation algorithms presented here is simple, it results in a significant 52% increase in recognition. More complex integration methods can be devised to further increase the recognition rate.

5 Conclusions and Future Work

In this work, we presented an unsupervised method to select and to detect landmarks in images coming from a video stream. Our results indicated that we achieved two main goals. First, we developed a robust working solution based on three steps: (i) selection of interesting image locations by means of a saliency algorithm, (ii) obtention of candidate landmarks using a multiple segmentation approach, and (iii) characterization of landmarks and their recognition by means of the SIFT algorithm and a database of previous detections. Second, we demonstrated that by using multiple segmentation algorithms, we can greatly increase the robustness of the system.

Our implementation showed a robust operation in real-world video streams without human supervision. The three main steps of the algorithm provided the necessary abstractions to obtain landmarks that were automatically selected and consistently detected in posterior images. Furthermore, the use of a saliency algorithm that emulates human physiology allowed us to obtain landmarks with a good correspondence with the underlying objects described by a human being.

It was shown that recognition results were in fact greatly influenced by the original segmentation utilized to generate a landmark. It was also demonstrated that segmentation algorithms can be very unstable and susceptible to worst-case scenarios, but that a mix of segmentations originating from different visual information can provide the necessary synergy to obtain adequate overall robustness. Hence our conclusion is that a multiple-segmentation approach can greatly increase landmark recognition.

As future work, new segmentation algorithms based on visual cues, such as depth or texture, can be added to the algorithm to increase its robustness.

Also, an important addition is to include learning schemes to determine a priori which segmentation algorithm may perform well in each case. If so, segmentation algorithms can be activated an deactivated dynamically to save computational cost and also to diminish the information actually stored in the database. On other hand, the saliency algorithm, which uses only bottom-up information from the input image, can be improved by means of top-down information.

Acknowledgments

This work was funded by FONDECYT grant 1050653. We would like to thank Luis Rueda and Anita Araneda for the valuable feedback in the preparation of this paper.

References

1. I. Biederman. Recognition by components: a theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.
2. J. Duncan. Integrated mechanisms of selective attention. *Current Opinion in Biology*, 7:255–261, 1997.
3. S. Harris. A combined corner and edge detector. pages 147–151, 1988.
4. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
5. N. Karlsson, L. Goncalves, M. Munich, and P. Pirjanian. The vslam algorithm for navigation in natural environments. *Korean Robotics Society Review*, 2(1):51–67, 2005.
6. D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
7. D. Mery and F. Pedreschi. Segmentation of colour food images using a robust algorithm. *Journal of Food engineering*, 66:353–360, 2004.
8. X. Munoz. *Image segmentation integrating colour, texture and boundary information.* PhD thesis, Universitat de Girona, 2002.
9. B. Schiele and J. Crowley. Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation*, 9:777–804, 1997.
10. A. Soto. *A Probabilistic Approach for the Adaptive Integration of Multiple Visual Cues Using an Agent Framework.* PhD thesis, Robotics Institute, Carnegie Mellon University, 2002.
11. Y. Sun and R. Fisher. Object-based visual attention for computer vision. *Artificial Intelligence*, 146:77–123, 2003.
12. J. K. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual-attention via selective tuning. *Artificial Intelligence*, 78(1-2):507–545, 1995.
13. D. Walther, U. Rutishauser, C. Koch, and P. Perona. Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision and Image Understanding*, 100:41–63, 2005.
14. D.R. Walther, D. Edgington and C. Koch. Detection and tracking of objects in underwater video. pages 544–549, 2004.

Hybrid Camera Surveillance System by Using Stereo Omni-directional System and Robust Human Detection

Kenji Iwata, Yutaka Satoh, Ikushi Yoda, and Katsuhiko Sakaue

National Institute of Advanced Industrial Science of Technology (AIST),
Information Research Technology Institute,
AIST Tsukuba Central 2, 1-1-1, Umezono, Tsukuba, Ibaraki, 305-8568, Japan
{kenji.iwata, yu.satou, i-yoda, k.sakaue}@aist.go.jp

Abstract. We propose a novel surveillance system that uses hybrid camera network. The system contains a Stereo Omni-directional System (SOS) and PTZ cameras to take the wide range images and face images of enough resolution for identification. The SOS can capture both omni-directional color images and range data simultaneously in real time. The robust human detection methods include robust background subtraction method (RRC), skin color segmentation and face tracking method. First, the system detects persons from an omni-directional image, and then the detailed face images are obtained with the PTZ cameras. The PTZ cameras can track the faces by using the four directional features and the relaxation matching. In addition, the system has automatic camera position calibration feature. Thus, the user can use the system without any troublesome settings.

Keywords: Surveillance, Stereo Omni-directional System, Hybrid Camera Network, Human Detection, Face Tracking.

1 Introduction

Rising consciousness of security have reinforced the needs of intelligent surveillance camera systems. Demands for surveillance cameras contain two conflicting purposes. One is observe wide range images at the same time, and another is to take high resolution face images for human identification. A fish-eye lens camera or an omni-directional camera is used to take the wide range images. However, these cameras cannot take images with enough resolution to perform facial recognition. On the other hand, pan-tilt-zoom (PTZ) camera can take the high resolution face image. Although, PTZ camera has a problem of flameout because the field of view is narrow. To address these problems, we integrate the two different types of cameras (the omni-directional camera and the PTZ camera) to achieve synergistic effect of them. We call this hybrid camera surveillance system.

Stereo Omni-directional System (SOS)[1] that has developed by the authors can capture omni-directional color and stereo images in real time with a full field of view of 360×360 degrees.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 611 – 620, 2006.

Vision-based person detection technology is very important for the surveillance systems[2]. Our person detection method includes a robust background subtraction method (RRC)[3], a skin color segmentation[4] and a relaxation matching method[5]. We realize the robust person detection and tracking feature by integrating these methods effectively.

In addition, the proposal system has an automatic camera position calibration feature. It is difficult for the users to calibrate multiple camera systems[6], but the system can be used without any troublesome settings.

2 System Configuration

2.1 Overview

A block diagram of the system is shown in Figure 1. Integrated control of the SOS and the PTZ camera is possible by means of a single PC.

A user arranges the PTZ cameras at arbitrary position. The system enters the automatic setup phase. In the phase, the positions of the PTZ cameras are automatically detected by the SOS.

In the surveillance phase, persons are detected from the SOS image. Regions of the persons are detected by integrating the background subtraction and skin color detection. 3-D positions of the persons can be determined by using omni-directional range data provided from the SOS. The PTZ cameras point the detected persons and track their movement by using robust face detection methods. In this way high resolution detailed face images are acquired.

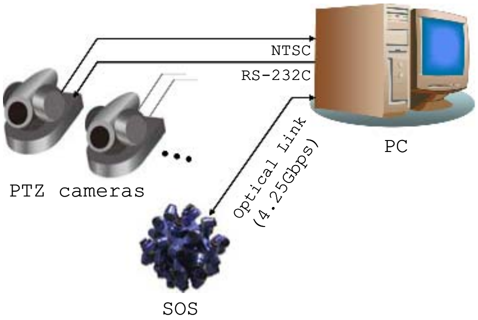


Fig. 1. Block diagram of the proposed system. 4.25 Gbps optical fibers connect the SOS and the PC. The images of PTZ cameras are captured by analog NTSC signals. Pan, tilt and zoom of the PTZ cameras are controlled from the PC via an RS232C connection.

2.2 Stereo Omni-directional System (SOS)

Figure 2 shows a photo of the external appearance of the SOS[1]. The SOS is a novel camera system capable of real-time, high-resolution capture of omni-directional color images and range information, without any blind spots, in an extremely compact design.

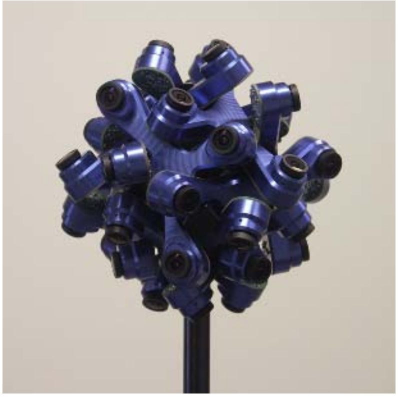


Fig. 2. Stereo Omni-directional System (SOS). 17cm diameter sphere consist of 36 CCD cameras.

The basic structure of the SOS consists of a regular dodecahedron (12 faces), with a three-camera stereo unit located on each face (making a total of 36 cameras). It is a problem that the size of a camerahead becomes excessively large when stereo camera units are arranged in such a regular-dodecahedron fashion. To address this problem, the three cameras of each stereo camera unit are mounted on a T-shaped arm, and by arranging the base planes of the stereo camera units so that they crisscross one another, we have ensured that and downsizing is possible while keeping a constant stereo baseline.

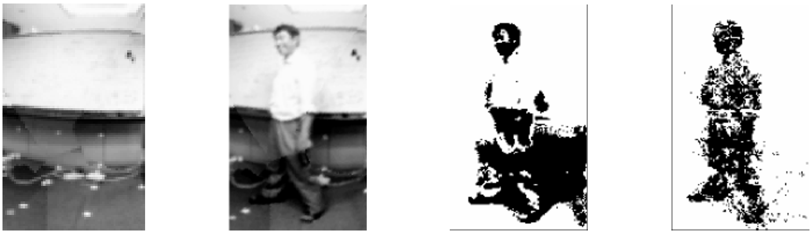
Each camera mounted to the stereo unit is on the same plane, and the optical axis of each is mutually parallel to the others. And the center camera is placed at right angles to the other two cameras so that their 75-mm base lines intersect at the center camera. In this way, each stereo unit satisfies the epipolar constraint; as a result, the processing cost of searching for corresponding points is decreased. Stereo calibration is performed for each stereo unit in terms of individual units, and the influence of lens distortion and misalignment between cameras is eliminated by software.

3 Object Detection by Using a SOS

This section describes the objects detection methods from omni-directional color and range images acquired by the SOS. The method consist the background subtraction by using RRC, color segmentation and range image processing.

4.1 Radial Reach Correlation (RRC)

Background subtraction algorithms are widely utilized as a technology for segmentation of background and target objects in images[7]. In particular, the simple background subtraction algorithm is used in many systems for its ease and low cost of implementation. However, because this algorithm relies only on the intensity difference, it has various problems, such as low tolerance for poor illumination and shadows and the inability to distinguish objects from their background when their intensities are similar. In an earlier study we proposed a new statistic, known as Radial Reach Correlation (RRC)[3], for distinguishing similar areas and dissimilar areas when comparing background images and target images at the pixel level. We achieved a robust background subtraction by evaluating the local texture in images. Figure 3 shows a comparison of background subtraction methods.



(a) Background (b) Scene (c) Simple subtraction (d) RRC subtraction

Fig. 3. Comparison of background subtraction method is shown. As for simple background subtraction (c), the region of a breast has not been detected because of its brightness distributions. Furthermore, shadow is strongly detected. On the other hand, the RRC image (d) has detected the person's region well. There are almost no influences by the shadows.

4.2 PTZ Camera Detection

The system has automatic camera position calibration feature. Thus, the user can use the system without any troublesome settings. The positions of the PTZ cameras seen from the SOS are obtained by the background subtraction by synchronizing with driving the mechanism of the PTZ cameras.

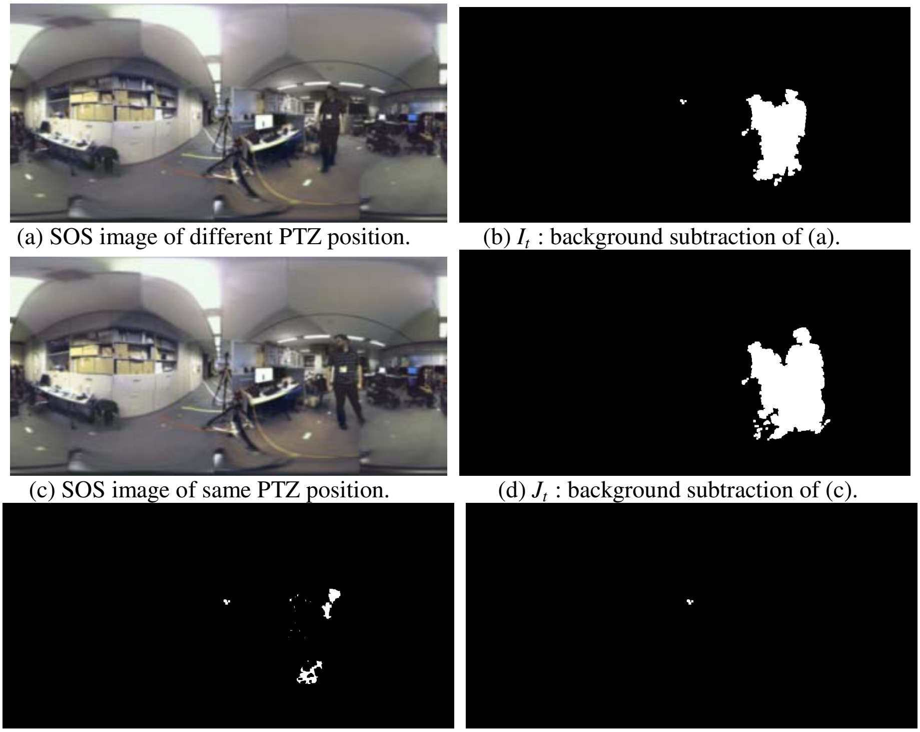
The detailed method of detecting one PTZ camera is described as follows. J_t express background subtraction by using RRC when pan and tilt are same position as background image. I_t express background subtraction when pan and tilt are different position as background image. t=1,2,3… is frequency of taking images. I_t and J_t are binary images. 0 means background, and 1 means foreground. Pixels in region of the PTZ camera in I_t are 1, in J_t are 0. Binary images K_t are defined as follows.

$$K_t = I_t \cdot \overline{J_t} \tag{1}$$

K_t include region of the PTZ camera and other moving objects such as person or noises. Then, other regions are reduced by using multiple images K_t .

$$L_t = K_t \cdot L_{t-1} \tag{2}$$

All pixels of image L_0 define 1. This operator repeats until connected region in L_t becomes one. The connected region is detected as the PTZ camera. 3D position of the



(a) SOS image of different PTZ position.

(b) I_t : background subtraction of (a).

(c) SOS image of same PTZ position.

(d) J_t : background subtraction of (c).

(e) K_t : candidate region of PTZ camera.

(f) L_t: detection result of PTZ camera.

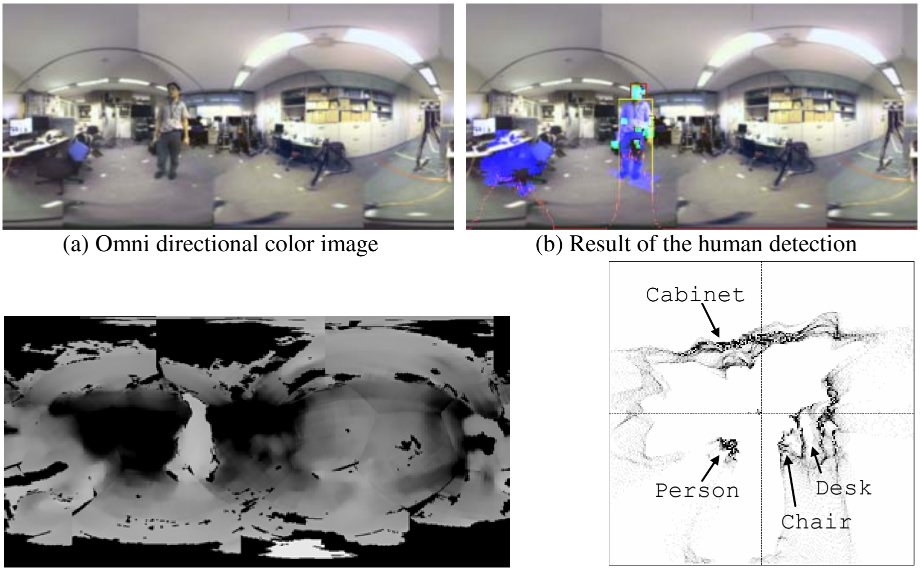
Fig. 3. Detection of a PTZ camera from SOS images. The PTZ camera was set up on a tripod of the center of the images. The position of PTZ camera can be detected by the logical operation of the subtraction regions. Figure (e) includes other moving objects. Figure (e) includes only PTZ camera.

PTZ camera is obtained by the range data from the SOS. Figure 4 shows an example of automatic detection of a PTZ camera from the SOS.

4.3 Human Detection

We use background subtraction, skin color detection and range segmentation for the human detection. Background subtraction uses RRC described in 4.1. Skin color detection uses normalized-RG color space. Skin color can be modeled very well by Gaussian distribution in normalized-RG color space [4].

The labeling is processed to the background subtraction regions. The labeled regions are segmented by using the histogram projected onto Y axis and X axis, because these regions include other than the persons. The labeled regions including the skin color is detected as the persons. The face regions are chosen from the skin regions by using simple geometry arrangement. Figure 5 shows an example of human detection.



(a) Omni directional color image

(b) Result of the human detection

(c) Omni directional range image

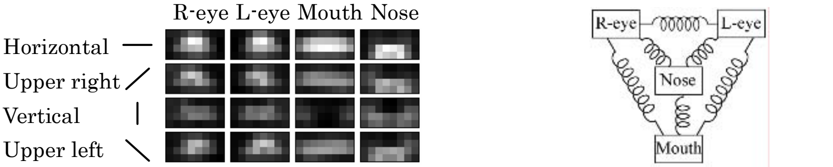
(d) Histogram projected onto the floor

Fig. 5. Process of human detection by using SOS is shown. Figure (b) and figure (c) are information captured by the SOS. Highlights in figure (b) are the regions of the background subtraction and the skin color detection. Rectangles in figure (b) are face and body detected as a person. Figure (d) shows the range information a histogram orthogonally projected onto the floor. It is clear from this histogram that 3D position of the person can be detected.

5 Face Tracking by PTZ Camera

The PC controls the PTZ cameras and acquires a high resolution face images from the face positions detected by the SOS. The pan-tilt-zoom cameras track the face by the color segmentation and facial parts detection.

We use template matching and relaxation method for the facial parts detection[5]. The templates need to be robust to changes in shape according to the face direction or individuals. Then, we use the four directional features[8]. The four directional features consist of horizontal, vertical, upper right, and lower right directional fields. Directional detection filters from a face image obtain four directional features. Figure 6(a) shows four directional features of the facial parts templates. These templates are the average of the four directional features of 10 persons facing in 7 directions. The



(a) Templates of facial parts

(b) Spring connection model of facial parts

Fig. 4. Facial parts detection by using template matching and relaxation method

template matching is difficult to pinpoint a facial parts position uniquely. Since the position of facial parts varies by individual or face direction, the concept of spring connection as shown in figure 6(b) is applied. We apply the relaxation operations[9] to facial parts detection by using the position relation of each face part.

The detailed method is described as follows. Initial probabilities $p_{ik}{}^{0}$ are evaluated with the similarities of the template matching by using the four directional features.

$$p_{ik}^{0} \propto \frac{\mathbf{T}_i \cdot \mathbf{I}_k}{\|\mathbf{T}_i\| \|\mathbf{I}_k\|} \;,\; \max_k(p_{ik}^{0}) = 1 \tag{3}$$

The four directional features of input image are expressed as $\mathbf{I}_k$. The templates are expressed as $\mathbf{T}_i$. i=1,···,4 express the right eye, the left eye, the mouth, and the nose.

The neighbor facial parts of the facial parts i are set to j=1,···,4($j \neq i$). Facial model defines relative position of facial parts. The relative position of the input is set to l. Neighbor probabilities $q_{jl}{}^{t}$ are maximum of the product of the Gaussian distribution to probability in the position l.

$$q_{jl}^{t} = \max_{l'}\left[p_{jl'}^{t} \exp\left(-\frac{d_{ll'}{}^{2}}{2\sigma^{2}} \right) \right] \tag{4}$$

l' is the whole input face domain. σ is the standard deviation of the Gaussian distribution. This is equivalent to the strength of a spring in the spring connection of the facial parts. $d_{ll'}$ is the distance between l and l'. Connection probabilities $r_{ik}{}^{t}$ are product of the neighbor probabilities $q_{jl}{}^{t}$.

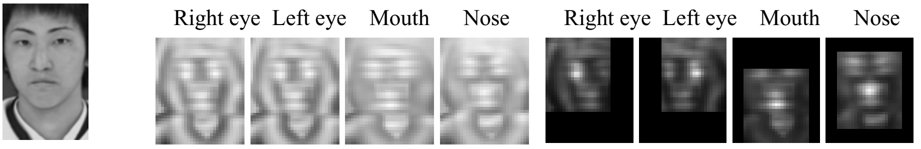
$$r_{ik}^{t} = \prod_{j} q_{jl}^{t} \tag{5}$$

Probabilities $p_{ik}{}^{t}$ are updated by using the connection probabilities $r_{ik}{}^{t}$.

$$p_{ik}^{t} \propto r_{ik}^{t-1} p_{ik}^{t-1} p_{ik}^{0} \;,\; \max_k(p_{ik}^{t}) = 1 \tag{6}$$

Two or more relaxation operations update probabilities $p_{ik}{}^{t}$. t shows the number of relaxation operation, which is t=0,1,···,τ. τ is repetition of the relaxation operation. τ=4 is used. Let the position k with maximum $p_{ik}{}^{t}$ be the position of the facial parts i. An example of the distributions of the probabilities is shown in figure 7.

This method can detect the face of the right and left 45 degrees and the top and bottom 30 degrees. If the probabilities of each facial part are more than threshold, the face is correctly detected, and controls the PTZ camera to locate the face at the center of the image. The initial zoom rate is set to obtain the standard size according to the distance to the face obtained with the SOS. The zoom rate is controlled at the following so that the face area obtained from the PTZ camera image may become the standard size.

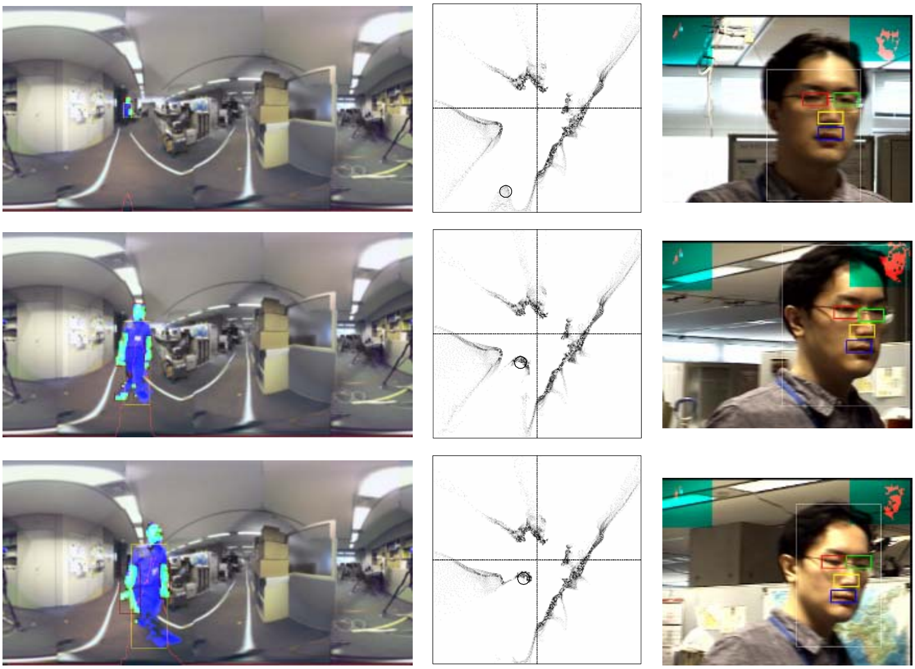


(a) Face region (b) Initial probabilities (c) Updated probabilities

Fig. 7. The distributions of the probabilities of the template matching to the face region of figure (a) are shown in figure (b). Whites indicate high probabilities. The correct positions of the facial parts are not clear in probabilities (b). Figure (c) shows the updated probabilities. The probabilities of correct positions are emphasized, and the probabilities of different positions are controlled. The correct positions are clear in probabilities (c).

5 Results

Example of setting up the system around the entrance of the office is shown in figure 8. A person approaches from the entrance. Figure 8(a) shows the result of detecting the person from the omni-directional color image acquired by the SOS. 3D position of



(a) Human detection from SOS (b) Position (c) Face tracking

Fig. 8. Results of human detection and face tracking. Highlights in figure (a) are the regions of the background subtraction and the skin color detection. Rectangles in figure (a) are face and body detected as a person. Figure (b) shows the depth information a histogram orthogonally projected onto the floor. Circles in figure (b) express the position of detected person. Figure (c) shows images obtained by PTZ camera. Rectangles in figure (c) are facial parts detected by relaxation matching method.

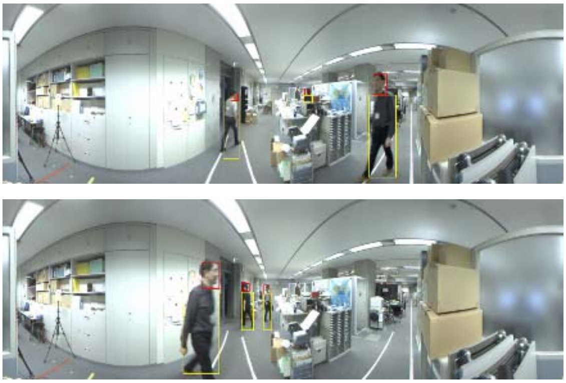


Fig. 9. Results of multiple people detection. Rectangles are face and body regions detected by the system.

the person is obtained by omni-directional range data. A PTZ camera points the detected person and tracks. Results of acquiring the face images are Figure 8(c).

This system works by not only one person but also multiple people. Figure 9 shows the examples of multiple people detection form the omni-directional color image. Thus, the system can watch the multiple people from the wide range at once time. Only one person can be taken in one PTZ camera, but the system is possible to deal with the multiple people by increasing the PTZ cameras.

6 Conclusion

We have proposed a hybrid camera surveillance system, which is integrating PTZ cameras and a SOS by using the robust human detection methods. We summarize the SOS, and describe the human detection methods by integrating the RRC, skin color detection and face tracking method. The experimental results showed the effectiveness of the proposed system. As future work, we would like to implement analysis of person actions.

References

1. H. Tanahashi, D. Shimada, K. Yamamoto and Y. Niwa: Acquisition of Three-Dimensional Information in a Real Environment by Using the Stereo Omnidirectional System (SOS), Proc. 3rd 3DIM, pp.365-371, 2001.
2. I.Haritaoglu, D.Harwood, and L.Davis: W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People, Proc. FG98, pp.222-227, 1998.
3. Y. Satoh, H. Tanahashi, C. Wang, S. Kaneko, S. Igarashi, Y. Niwa and K. Yamamoto: Robust Event Detection by Radial Reach Filter (RRF), Proc. ICPR2002, Vol.II, pp.623-626, 2002.
4. Jean-Christophe Terrillon, Martin David and Shigeru Akamatsu: Automatic Detection of Human Faces in Natural Scene Images by Use of a Skin Color Model and of Invariant Moments, Proc. FG98, pp.112-117, 1998

5. K. Iwata, H. Hongo, Y Niwa, and K. Yamamoto: Robust Facial Parts Detection by Using Four Directional Features and Relaxation Matching, Proc. KES2003, LNAI2774, pp.882-889, 2003.
6. Xilin Chen, Jie Yang and Alex Waibel: Calibration of a Hybrid Camera Network, Proc. ICCV2003, Vol.1, pp.150-155, Oct. 2003
7. Q. Zang and R. Klette : Robust Background Subtraction and Maintenance, Proc. ICPR2004, Vol. 2, pp.90-93, 2004.
8. K. Yamamoto: Present State of Recognition Method on Consideration of Neighbor Points and Its Ability in Common Database, Trans. IEICE, Vol.E79-D, no.5, pp.417-422, 1996
9. A. Rosenfeld, R. Hummel, S. Zucker: Scene Labeling by Relaxation Operations, IEEE Trans. Sys. Man and Cybern., SMC-6, pp.420-433, 1976

Robust TV News Story Identification Via Visual Characteristics of Anchorperson Scenes

Chia-Hung Yeh[1], Min-Kuan Chang[2], Ko-Yen Lu[2], and Maverick Shih[3,*]

[1] Department of Computer Science and Information Engineering,
National Dong-Hwa University 974, Taiwan
[2] Department of Electrical Engineering, National Chung-Hsing University 402, Taiwan
[3] MAVs Laboratory Incorporated, Lung-Tan, Tau-Yuan 325, Taiwan
yeh@mail.ndhu.edu.tw, minkuanc@dragon.nchu.edu.tw,
bencoper.tw@yahoo.com.tw

Abstract. In this paper, a new scheme for TV news segmentation via exploring the efficient visual features is proposed especially for TV news which contains lots of changeful background of anchorperson shots. The proposed scheme can be divided into two parts: probable anchorperson shot detection and real anchorperson detection. Our proposed method can efficiently detect anchorperson shots even though anchorperson shots contain changeful background and anchorperson position variation. Meanwhile, non-anchorperson shots can be robustly excluded from report shots such as interview scenes. Experimental results are given to demonstrate the feasibility and efficiency of the proposed techniques.

Keywords: TV news segmentation, visual features, skin color, face recognition, anchorperson detection.

1 Introduction

Because of the rapid development of multimedia techniques in recent years, a large amount of information can be easily obtained. Although they offer many conveniences to our daily life, the ways to select the needs from the numerous growing multimedia data in fast way is difficult. Therefore, multimedia data indexing has already become an important research topic [1][2][3]. For various kinds of data types such as text, image, audio and video, video is the most complicated one, since it combines several data types into that. In the real world, video genres include movies, news, and current affairs, sports, commercials and etc. In this paper, our focus is on TV news structure analysis. Due to video length and unstructured format, efficient access video is not easy. If we index the video by the artificial way, it will take a lot of time. For this reason, we need to develop automatic TV news video indexing method.

Due to huge amount of TV news happening everyday, such as social, entertainment, politics, sports news, and etc., it is important to develop automatic techniques for efficiently indexing, browsing and searching TV news videos. As we know, TV news is composed of several stories which are introduced by anchorperson and each

[*] Corresponding author.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 621 – 630, 2006.

story is organized of anchorperson and reports' shots shown in Fig. 1. TV news segmentation is an essential preprocessing step for further proposes such as search and classification. When searching interesting stories in new videos, artificially browsing through news segments is usually impractical and time consuming. Therefore, fully automatic TV news segmentation system is important for indexing, browsing and searching TV news.

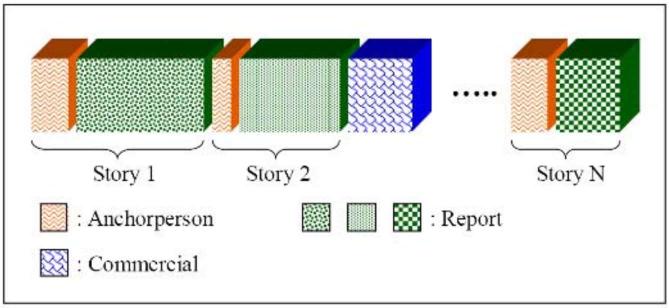


Fig. 1. The Structure of TV news videos

In recent years, many research related to TV news segmentation have been published. We will discuss the several representative research as follows. Neil O'Hare [4] proposed a scheme to segment TV news by SVM (support vector machine). The feature of their work is to train SVM, and the trained SVM classifier can be used to detect the anchorperson shots within TV news videos. N. O'Connor [5] proposed a hierarchical visual index structure for automatically segmenting TV news stories. The shot clustering is the core component in this approach. Hui [6] exploited the visual information for story boundary detection and included several techniques such as spatial difference metric (SDM), histogram difference metric (HDM), fuzzy c-means (FCM) and graph-theoretical clustering (GTC). In the previous TV news segmentation systems, shot change can be detected accurately by color histogram difference due to the assumption of the static background of anchorperson shots. However, this assumption is inappropriate for several countries' TV news due to the changeful background of anchorperson shots. In addition, the procedures of the above systems are complicated and their systems include many techniques which need a large amount of computational complexity.

In this paper, we propose a new approach to improve efficiency and precision of the news story segmentation system. The proposed scheme employs the skin color detection to roughly determine the probable anchor shots because anchor shots contain the face region. Then, we use the specific shot change detection, time constraint method and region histogram based on non-skin color method to exclude the report shots in order to extract the real anchor shots. The proposed method can significantly reduce the computational complexity and is efficient for TV news videos which have changeful background of anchorperson shots and anchorperson position variation. Simulation results are given to demonstrate the feasibility and efficiency of the proposed technique. The rest of this paper is organized as follows. Section 2 describes the proposed TV news story segmentation system that includes several key techniques. Experimental results are given in Section 3. Finally, concluding remarks and some future research directions are given in Section 4.

2 Proposed TV News Story Segmentation System

In order to divide TV news into individual story units, the boundary between story units should be detected accurately. Figure 2 shows the flowchart of the proposed TV news segmentation system. In our proposed approach, we exploit the anchorperson shot's features and the characteristics of TV news structure to detect story boundaries. Each story is composed of two kinds of shots: an anchorperson shot and a report shot. If we can successfully detect anchorperson shots, TV news can be split into many individual stories. The followings are the explanation of the key components of the proposed system that can be roughly separated into two parts: probable anchorperson shot determination and real anchorperson shot determination.

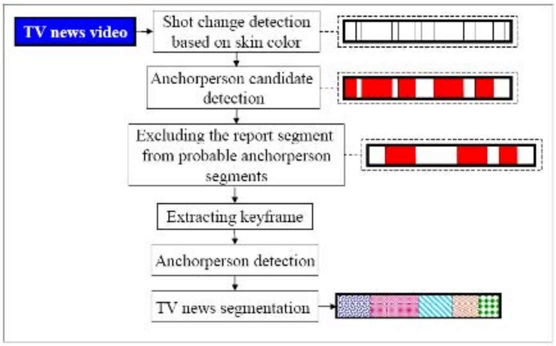

TV news video
Shot change detection based on skin color
Anchorperson candidate detection
Excluding the report segment from probable anchorperson segments
Extracting keyframe
Anchorperson detection
TV news segmentation

Fig. 2. Key components of the proposed news story segmentation system

2.1 Probable Anchorperson Shot Determination

2.1.1 Shot Change Detection Based on Skin Color

Shot change detection is usually the first step to do any video content analysis [4][5][6][7][8]. Shot consists of a set of consecutive and similar frames. Extensive work has been done in this area. A simple way to achieve shot change detection is to

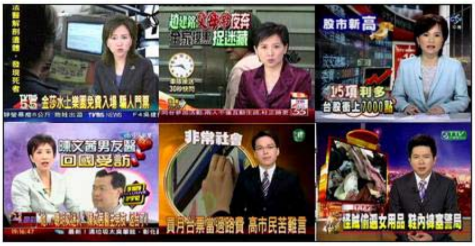


Fig. 3. Large variety of Nowadays TV news

employ the color histogram difference. Conventional shot change detection method cannot accurately split TV news into anchorperson and report segments due to the large variety of Nowadays TV news shown in Fig. 3.

In this work, we exploit the efficient visual features of anchorperson for detecting anchorperson shot change shown in Fig. 4. An anchorperson shot surely contains face region, and this feature is informative and important for anchorperson shot change detection. If conventional face detection algorithms are employed to find face regions out in TV news videos [4][9][10][11], it will cost large amounts of computational complexity and it is impractical in the real-time applications. In order to reduce computational complexity of face detection, we only use skin color information as the major tool for searching face region. In general, the computational complexity of skin color detection should be lower than that of face detection algorithms, but the skin color detection is sensitive to the conditions of lighting and the object that the color is similar to skin color. Therefore, after discovering the skin-color parts in videos, some parts exist with lots of noise regions shown in Fig. 5.

We can utilize the grouping method to get the section that contains the face in each frame and exclude noise regions shown in Fig. 5. Meanwhile, slight changes exist in the position of anchor's face during anchorperson shots, and the time duration of an anchorperson shots should be shorter than that of a report shot. As a result, we make use of the properties above called time constraint method to locate the probable anchorperson segments shown in Fig. 6. In the next section, we will explain the ways to exclude report shots from probable anchorperson shots.

2.1.2 Exclusion of Report Shots from Probable Anchorperson Shots

The probable anchorperson shots might involve report shots such as interview scenes. Here, we extend anchorperson face regions along the *y*-direction to cover the certain non-face region that contains the anchor's suit information. This feature is so informative because an anchorperson usually dresses the same suit during broadcasting news. We call this method as region histogram based on non-skin color to detect report shots shown in Fig. 7. In Fig. 7, the distribution of histogram of this non-skin color

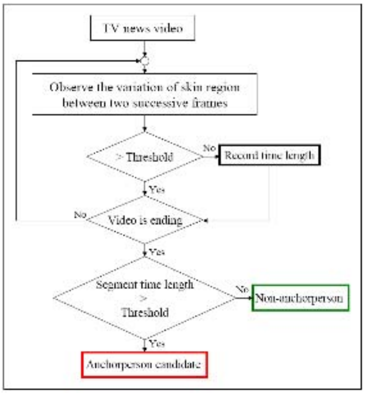


Fig. 4. Shot change detection based on skin color

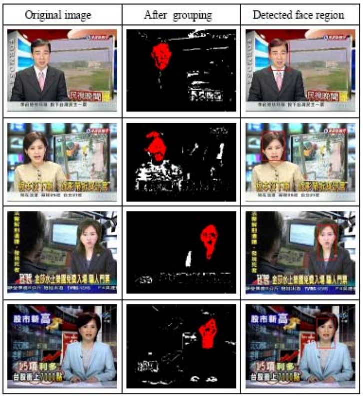


Fig. 5. Examples for grouping method

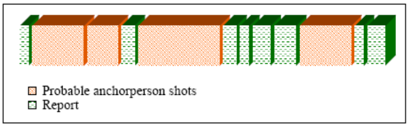


Fig. 6. Probable anchorperson shots

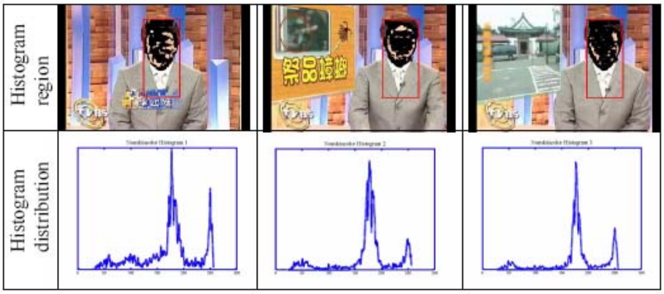


Fig. 7. Region histogram based on non-skin color

region is very similar; and this feature can be used in the self-recognizing algorithm that will be discussed in the next section to exclude report shots from the probable anchorperson shots.

2.2 Real Anchorperson Shot Determination

2.2.1 Self-recognizing Algorithm

In TV news video, anchorperson shots surely own the same features; for example, an anchorperson clothes should be the same in these anchorperson shots. This feature can be represented by the non-skin color region histogram information. The proposed method is called the self-recognizing, which exploits the correlation among real anchorperson shots to select a keyframe and use this keyframe to validate all probable anchorperson shots. We first determine the keyframe from each shot by simply selecting first frame in a shot and then calculate the skin-color region histogram differences with the other keyframes and accumulate them. When all keyframes have their own sum of differences shown in Table 1, the summation value represents the similarity among each shot. If the summation value for each shot is relatively larger than others, this shot will be deleted. This means that the deleted shot is not similar to the others and it maybe belong to a report shot.

Table 1. Self-recognizing algorithm

Difference	Shot1	Shot2	Shot3	Shot4
Shot1	0	Difference(2-1)	Difference(3-1)	Difference(4-1)
Shot2	Difference(1-2)	0	Difference(3-2)	Difference(4-2)
Shot3	Difference(1-3)	Difference(2-3)	0	Difference(4-3)
Shot4	Difference(1-4)	Difference(2-4)	Difference(3-4)	0
$\sum Diff_i$	$\sum Diff_1$	$\sum Diff_2$	$\sum Diff_3$	$\sum Diff_4$

2.2.2 Modified Pattern Method

In the self-recognizing algorithm, the skin-color region may be polluted by light or skin-like color, the histogram produced by region histogram based on skin-color method will be interfered. In order to improve this interference, the projection method is used to modify the selected region. The approach is to scan the image column-by-column and row-by-row shown in Fig. 8. The result of the projection method represents the skin-color distribution of *y*-axis and *x*-axis, which can be exploited to process the skin-color pattern in a frame. In addition, the human face region obtains an obvious feature in which the height of skin-color pattern is longer than the width. Be

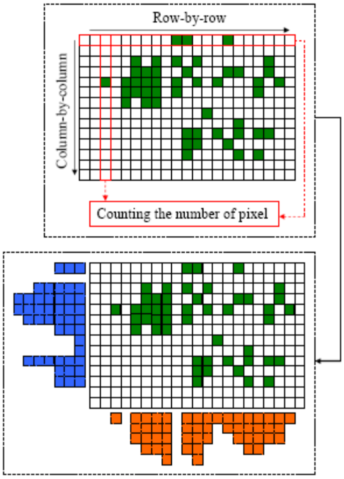


Fig. 8. Illustration of projection method

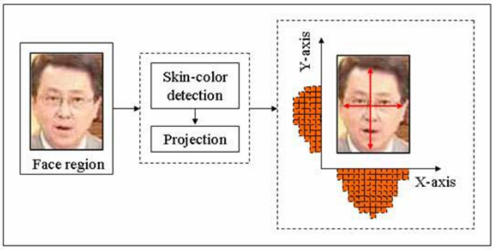


Fig. 9. Distribution of the *x*-axis and the *y*-axis

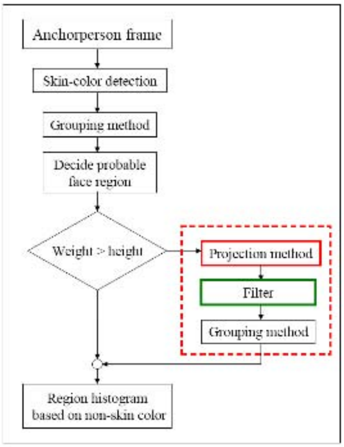


Fig. 10. Flowchart of modified pattern algorithm

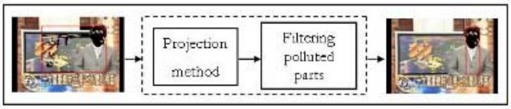


Fig. 11. The detection result

sides, the face part in the skin-color region is usually larger than others, and the *x*-axis of the face region contains more information than *y*-axis shown in Fig. 9. As a result, in order to cost down the computational complexity, we only project the skin-color of a frame to *x*-axis. Thus, if the width is longer than height, the pattern will be modified by the projection method shown in Fig.10. After filtering the polluted parts in the probable face region, the non-skin color region histogram method can obtains the information of anchorperson clothes accurately shown in Fig. 11. We will give a summary for anchorperson shot detection method in the next section.

2.2.3 Summary of Anchorperson Shot Detection

According to the TV news structure, a news story includes two shots: anchorperson shots and report shots. An anchorperson introduces TV news before report shots in the news story. If anchorperson shots are identified, news story segmentation is a straightforward process. We can detect anchorperson in TV news video with the following method. We first extract a keyframe from a probable anchorperson shot that obtained from Section 2.1 to represent this shot. Then, the modified pattern algorithm is used to process keyframes and this procedure will improve the interference for region-histogram. The process of setting range of region-histogram method is based on the probable face region, and we make use of this region to search the non-skin color region. The selected region includes the anchorperson clothes and anchorperson face. If we only utilize the anchorperson face to detect anchorperson, the result will be inaccurate. This is because the color histogram of human face is similar to each other, and the report shot has the probability to contain the face pattern such as interview scenes. However, the anchorperson's clothes surely have no change and the color of anchorperson's clothes is different from interviewer. According to above discussing, the color of anchorperson's clothes is very informative for real anchorperson shot detection. Thus, we extract the color of anchorperson's clothes by the region-histogram method based on non-skin color in each keyframe. To this end, the self-recognizing is used to detect the anchorperson based on the color of anchorperson's clothes in TV news video frame-by-frame and the flowchart of the above method shown in Fig. 12.

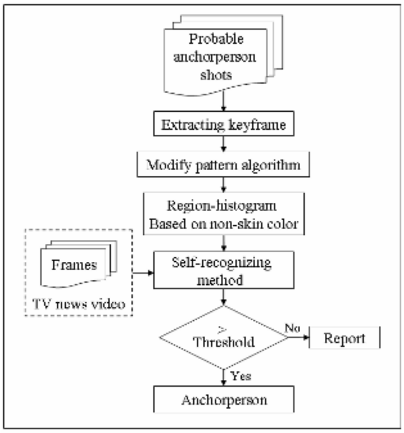


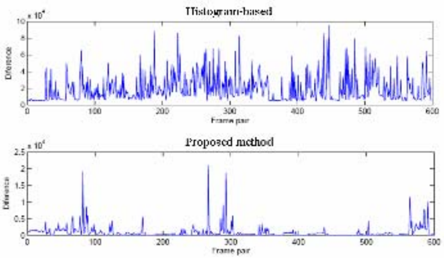
Fig. 12. Real anchorperson shot detection by region-histogram

3 Experimental Results

We evaluate the proposed system with five kinds of TV news video whose length are 10 minutes. The tested videos belong to MPEG-2 format with a frame size of 352 x 240 pixels and a frame rate of 30 bps.

3.1 Results of Shot Change Detection Based on Skin Color

When we apply the conventional shot change detection on TV news videos, the detection precision is inaccurate due to the changeful background. In the anchorperson shot, the obvious feature is the skin color, and we exploit this feature to detect the shot change detection. If we can detect the anchorperson shot successfully, the detect precision of news story segmentation will be accurate. Figure 13 shows shot change detection results between conventional histogram-based method and the proposed skin-color based method. As can be seen in Fig. 13, the proposed shot change detection make significant improvement compared to the conventional method.



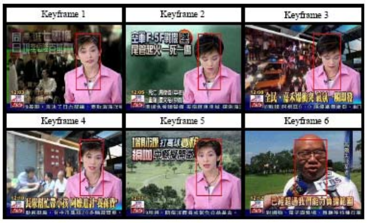
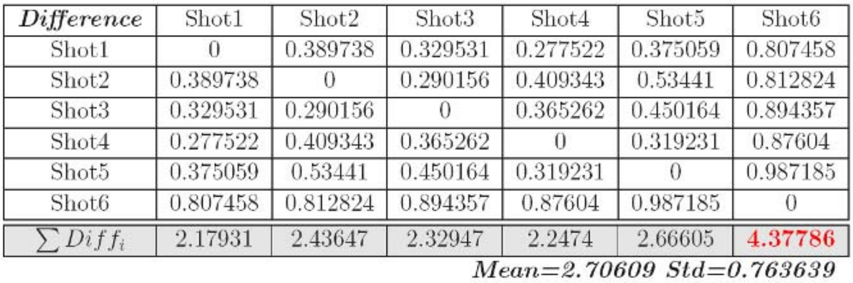


Fig. 13. Results of different shot change detection methods **Fig. 14.** Selected Keyframes

3.2 Evaluation of Proposed System

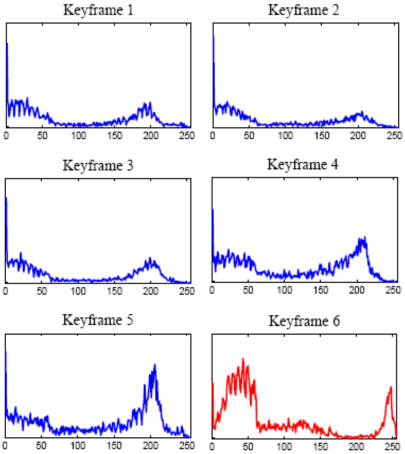
After shot change detection based on skin color, we use the time constraint to obtain probable anchorperson shots from news video, then we extract the keyframes from probable anchorperson shots to represent each shot. The extracted keyframes are shown in Fig. 14. However, in the probable anchorperson shots may contains the report shots such as shot 6, the self-recognizing algorithm can be applied to exclude report shots in probable anchorperson shots. Table 2 shows that results of the self-recognizing algorithm. Via this process, we can easily exclude shot 6 that is not the real anchorperson shot because the region histogram based on non-skin color of the shot 6 is not similar to others shown in Fig. 15. Figure 16 shows that the exclusion of non-anchorperson shot. Finally, TV news can be indexed by anchorperson on report shots for further applications such as browsing, searching, and retrieval. Two measurements, recall and precision, are used to evaluate the performance of the proposed scheme. They are measured against the groundtruth in Table 3. For most of the tested TV news videos, our algorithm can get the high accuracy for TV news segmentation. It is worthwhile to mention that the result for SETN is not good because it contains a lot of interview scenes, and the performance is expected to improve if we fine-tune the algorithm.

Table 2. Illustration of self-recognizing algorithm



Difference	Shot1	Shot2	Shot3	Shot4	Shot5	Shot6
Shot1	0	0.389738	0.329531	0.277522	0.375059	0.807458
Shot2	0.389738	0	0.290156	0.409343	0.53441	0.812824
Shot3	0.329531	0.290156	0	0.365262	0.450164	0.894357
Shot4	0.277522	0.409343	0.365262	0	0.319231	0.87604
Shot5	0.375059	0.53441	0.450164	0.319231	0	0.987185
Shot6	0.807458	0.812824	0.894357	0.87604	0.987185	0
$\sum Diff_i$	2.17931	2.43647	2.32947	2.2474	2.66605	**4.37786**

Mean=2.70609 Std=0.763639



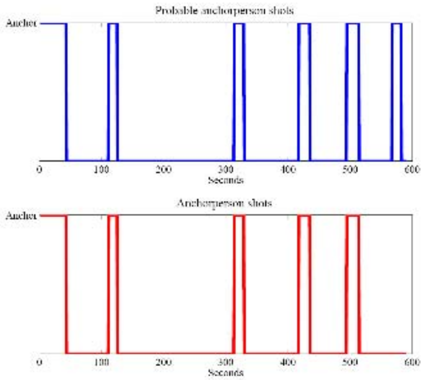
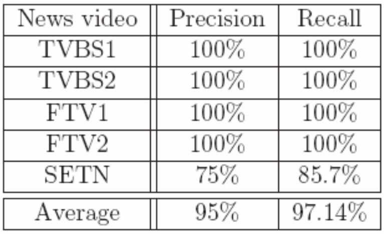


Fig. 15. Region histogram based on non-skin color **Fig. 16.** Exclusion of report shots

Table 3. Performance of anchorperson shot detection



News video	Precision	Recall
TVBS1	100%	100%
TVBS2	100%	100%
FTV1	100%	100%
FTV2	100%	100%
SETN	75%	85.7%
Average	95%	97.14%

4 Conclusion

In this paper, we deeply discussed the visual characteristics for TV news segmentation, and successfully indexed the anchorperson shots. Therefore, we can segment TV news into individual story units for further applications such as indexing, browsing and searching. Our proposed algorithm employed skin color detection, time constraint, and region histogram in non-skin-color region methods instead of conventional face detection methods. According to the experimental results, our proposed method can significantly reduce the computational complexity and the detection precision of TV news segmentation is really high. After segmentation, the TV news video can be split into many individual stories, and the summarization techniques can be used to obtain the highlight of each news story or achieve news classification for more applications.

References

1. C.-H. Yeh, S.-H. Lee & C.-C. Jay Kuo, "Content-based video analysis for knowledge discovery," *Handbook of Pattern Recognition and Computer Vision 3rd Edition Version*, Editor by Prof. C. H. Chen and Prof. P.S.P. Wang, World Scientific Publishing Co. ISBN: 981-256-105-6.
2. Y. Li, S.-H. Lee, C.-H. Yeh & C.-C. Jay Kuo, "Techniques for movie content analysis and skimming," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 79-89, 2006.
3. S.-H. Lee, C.-H. Yeh & C.-C. Jay Kuo, "Automatic movie skimming with story units via general tempo analysis," in *Proceedings of SPIE Electronic Image Storage and Retrieval Methods and Applications for Multimedia*, vol. 5307, pp. 396-407, 2004.
4. N. O'Hare, A. F. Smeaton, C. Czirjek, N. O'Connor & N. Murphy, "A generic news story segmentation system and its evaluation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 1028-31, 2004.
5. N. O'Connor, C. Czirjek, S. Deasy, S. Marlow, N. Murphy & A. Smeaton, "News story segmentation in the Fischlar video indexing system," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, pp. 418-421, 2001.
6. P. Y. Hui, X. Tang, H. M. Meng, W. Lam & X. Gao, "Automatic story segmentation for spoken document retrieval," in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems,* vol. 3, pp. 1319-1322, 2001.
7. M. M. Yeung, B.-L. Yeo & B. Liu, "Extracting story units from long programs for video browsing and navigation," in *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems,* pp. 296 -305, 1996.
8. Y. Rui, T. S. Huang & S. Mehrotra, "Constructing Table-of-Content for Videos," in *Proceedings of the ACM Journal on the Multimedia Systems*, vol. 7, no. 5, pp. 359-368, 1999.
9. A. Albiol, C. A. Bouman & E. J. Delp, "Face detection for pseudo-semantic labeling in video databases," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, pp. 607-611, 1999.
10. C. Czirjek, N. O'Connor, S. Marlow & N. Murphy, "Face detection and clustering for video indexing applications," in *Proceedings of Advanced Concepts for Intelligent Vision Systems*, 2003.
11. J. José de Dios, & N. García, "Fast face segmentation in component color space," in *Proceedings of the IEEE International Conference on Image Processing*," vol. 1, pp. 191-194, 2004.

Real-Time Automatic Calibration for Omni-display in Ubiquitous Computing

Dongwuk Kyoung, Yunli Lee, Eunjung Han, and Keechul Jung

HCI Lab., School of Media, College of Information Science, Soongsil University,
Seoul, South Korea
{kiki227, yunli, hanej, kcjung}@ssu.ac.kr
http://hci.ssu.ac.kr/

Abstract. In recent years, projectors are undergoing a transformation as they evolve from static output devices to portable, or communication systems. However, the projection images appear distorted unless the projector is precisely aligned to the projection screen. Generally many projection-based systems are corrected for oblique projection distortion using calibration methods (e.g., warping function). Computing a warping function uses fiducials or a special pattern projected to the screen. The methods can not use automatic calibration in real-time, like when projector is moving. We introduce a new technique for a real-time automatic calibration on planar surfaces using projected corner points. Our system provides a function of correcting image during the movement of the projector. An advantage of the proposed method is the system can be easily applied to ubiquitous computing.

Keywords: ubiquitous display, real-time calibration, camera-projector systems, computer vision.

1 Introduction

Ubiquitous computing envisions the world for possibility to access computer resources data at anywhere and anytime, and the services are available through the Internet [1]. Most of Internet data is designed to be accessed through a high-resolution graphical interface. This means possibility exist of carrying displays. The displays for ubiquitous computing (ubiquitous displays) are a projection-based system, because it offers new revolutionary possibilities for display, with opportunistic projection onto nearby surfaces like walls and tabletops to create a display wherever needed. However, the system is required to correct oblique projection by geometric calibration.

Many different approaches to ubiquitous displays [2] which are Display Walls, Steerable Projector Camera Systems and Mobile Projectors are being investigated, ranging from display walls and steerable projectors to new display materials. There are many approaches to address geometric calibration. Table 1 shows an overview of the calibration method for ubiquitous displays.

Display Walls (i.e., Rear-Projected Display and Front Projected Display Walls) appear to offer large scale high resolution display, a solution to the small display area problem. However, these systems are not suitable for the future office as the tiled

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 631 – 640, 2006.

display systems are employ to take up a large floor space, time consuming for calibration and there are fixed, not portable because the projected pattern image is needed for calculating calibration method of the Display Wall [3-8].

Steerable Projector Systems surfaces can dynamically change if its projection becomes occluded and also creates the potential for novel display types such as user following displays [9-12]. Everywhere Display[1] in an environment is developed by Pinhanez [9]. This system is to couple an LCD/DLP projector to a motorized rotating mirror and to a computer graphics system that can correct the distortion caused by oblique projection. The Steerable Projector System can not correct the distortion in real-time, because the calibration parameters of the virtual 3D surface are determined manually by simply projecting a special pattern and interactively adjusting the scale, rotation, and position of the virtual surface in the 3D world.

Table 1. A Comparison of Display Technologies for Ubiquitous Display

Display	Display Technology	Mobility	Image Calibration Method		
			Calculation Method	**Device Requirement**	**Processing Time**
Traditional Computer Monitors	CRT and LCD	Fixed	-	-	-
Rear-Projected Display Walls [3-5]	Projection	Fixed	Off-line (automatic)	With Camera	Not Real-Time
Front Projected Display Walls [6-8]	Projection	Mobile, ad-hoc creation	Off-line (automatic)	With Camera	Not Real-Time
Steerable Projected Display Walls [9-12]	Projection	Fixed but steerable image	Off-line (manual or semi-automatic)	With Camera and Marker	Not Real-Time
Mobile Projectors [13-14]	Projection	Portable	On-line (automatic)	With Camera and Marker	Real-Time
Our System	**Projection**	**Portable or steerable image**	**On-line (automatic)**	**With Camera**	**Real-Time**

Mobile Projector System [13-14] developing a mobile projector-camera system that fit into a large toolbox. Despite this achievement the system still relied on a connected mains cable for powering the video projector. However, the promise of Laser and LED projectors that can run for hours on batteries heralds a new era in mobile display technology. Raskar [13] describes object augmentation using a hand-held projector[2]. This system does object recognition by mean of markers attached to the object of interest. The markers are “piecodes”, colored segmented circles. These markers are used to compute camera pose (location and orientation) and hence the oblique projection is calibrated. Mobile Projector is determined automatically in

[1] Steerable Projector Camera System was initially developed by Pinhanez at IBM.

[2] Raskar at the Mitsubishi Electric Research Labs (MERL) popularized the idea of handheld projector-camera system.

real-time during a change of location and orientation of projector by marker because the marker is used to compute camera pose (location and orientation) and hence the oblique projection is calibrated. However, this system can not calibrate in projection space which is not attached marker. Generally, many calibration methods for oblique projection need camera and projector pose (location and orientation). However, the existing methods can not automatically calibrate in real-time, like when projector is moving.

In this paper, we proposed an automatic calibration system in real-time without attach fiducials to the screen. Our idea is to project images with 4 corner points instead of attach fiducials to the screen, and the system automatically pre-warps the image to be displayed resulting in a perfectly aligned and rectilinear image. Therefore, different surfaces become available to be used as displays, when the motion of the projector. An advantage of the proposed method is that it can be easily applied to ubiquitous computing.

2 Projector-Based System for Real-Time Automatic Calibration

The projector-based system needs distortion information for real-time automatic calibration. Our system uses projection markers for distortion information. The idea of our system is to project image with 4 corner points, and the computer vision algorithms use for detecting corner points (Fig. 1).

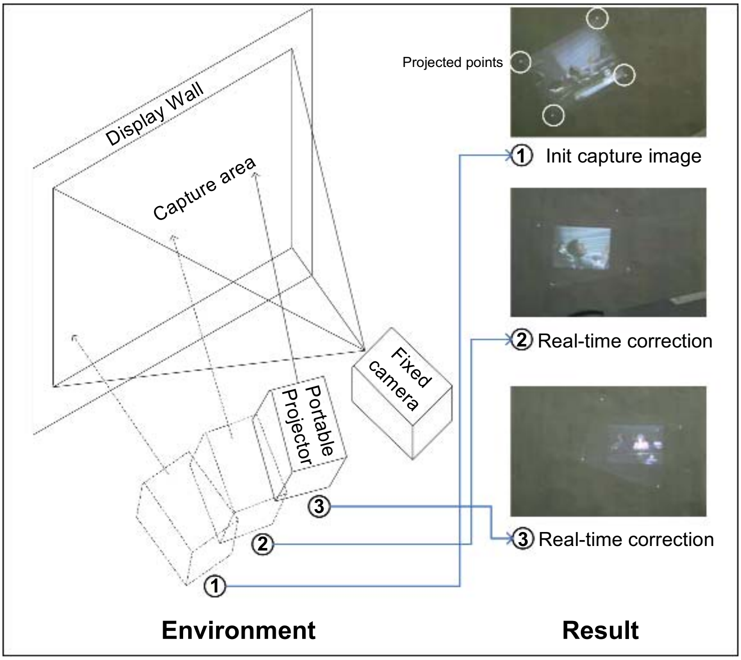


Fig. 1. Overview of our system

2.1 Overall System Architecture

The real-time geometric calibration system is composed of a DLP projector and an inexpensive web camera. The web camera has been used as a fixed device but the projector has been used as an unfixed device. The projector is connected to the display output of a computer that performs geometric calibration. The Fig. 2 shows the architecture of a proposed system.

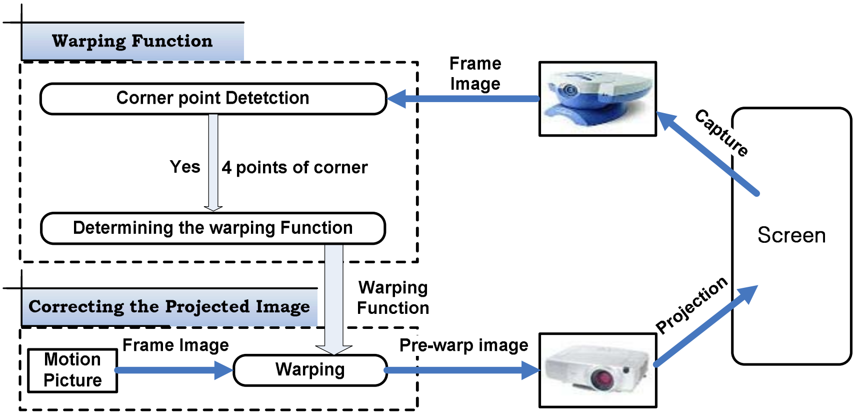


Fig. 2. The real-time geometric calibration system architecture

The automatic calibration system corrects oblique projection with a camera, a projector, and a calibration application. This system takes 4 steps. The first step is projection a frame image of a motion picture by a projector. The second step is capturing a projection image by an inexpensive web camera. The third step computes warping function with corner detection process. The last step applies the warping function to a frame image of motion picture.

2.2 Detection of Corner Points

Detection of corner point steps for to find 4 corner points takes 6 steps.

1. Converting from the source image with 4 corner points to binary image.
2. Filter the binary image to remove noise.
3. Find components (points) by using connected component labeling method.
4. Calculate center coordinate of each components calculate by using average of pixel consist of components.
5. Get inner square by using center coordinate.
6. Find corner points by using diagonal of corner square.

The camera captures both 4 corner points and general (presentation) image (Fig. 3(b)). In this image, 4 corner points for calculate warping function is determined. As previous work for calculate warping function, component, set of points, is found after execute binary image and morphology operation (Fig. 3(c)). The binary image eliminated unnecessary information on image except the 4 corner points. We generate the binary image using threshold which is average of RGB value determined by experiment result.

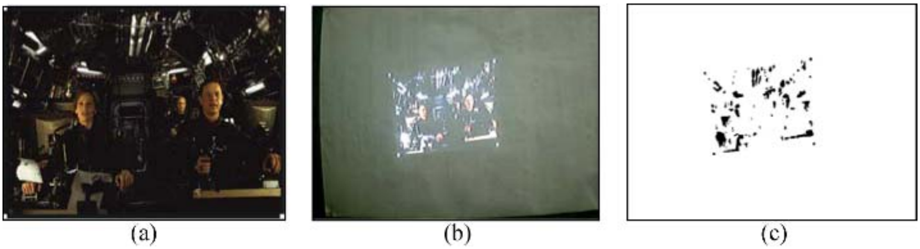


Fig. 3. Captured image processing: (a) initial projected image (marked 4 corner points), (b) captured image, and (c) result image after binary and morphology operation

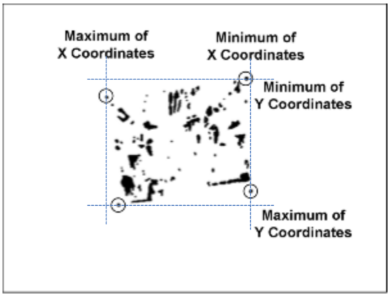


Fig. 4. Maximum and Minimum of X and Y coordinates

After morphology operation, center coordinate at each component (average coordinate of pixel compositing component) is used representation coordinate of component. Maximum and minimum components (4 corner component) are found by sorting according to x value and y value (Fig. 4).

Then we can draw corner square including 4 corner components (Fig. 5). The components meeting corner square are called corner points and 2, 3 or 4 corner points are found according to the shape of the inner square made by connect corner points. But when inner square is rectangular form (include a perfect square), we except that case in the Fig. 5 because that case is same case finding 2 corner point.

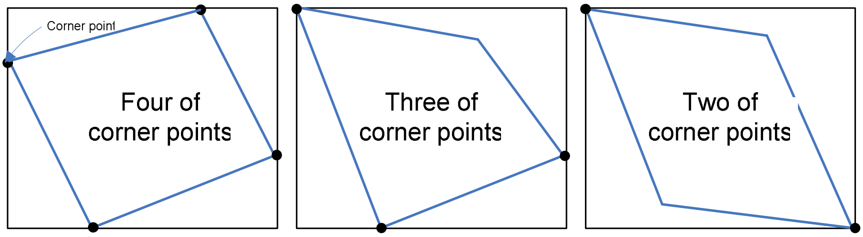


Fig. 5. Corner points according to the shape of square

Fig. 6 (a), (b) method use diagonal of corner square. When we draw vertical from each diagonal to all of component, we can find 4 components having longest vertical. Because we need coordinate information of each component for calculate warping function in a next step, we store coordinate information of each component.

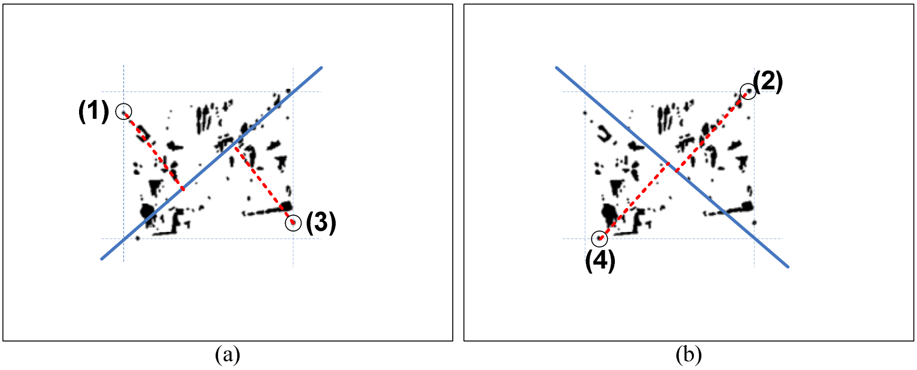


Fig. 6. Finding corner square and corner points using coordinate sorting

2.3 Warping Function Determination

The warping function is determined using keystone correction method and four 4 corner points [6]. Fig. 7 summarizes relationships between the frames of reference that are relevant to the problem of automatic keystone correction. The relationships between the three frames of reference corresponding to the source image frame, camera image frame and projected image frame (Fig. 7(a)). The application image can be appropriately pre-warped, using the mapping W so that it appears rectilinear after projection through a misaligned projector (Fig. 7(b)).

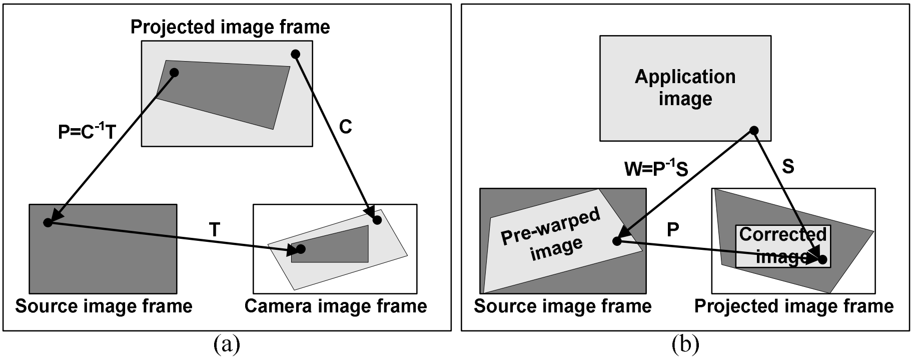


Fig. 7. Relationships between the frames

The keystone correction establish the mapping (T) between projector and camera, and the mapping (C) between screen and camera (like the equation 1).

The mapping (P) between projector and screen is defined as the equation 2.

$$T \times \begin{pmatrix} x_{proj} \\ y_{proj} \\ 1 \end{pmatrix} \equiv \begin{pmatrix} x_{cam} \\ y_{cam} \\ 1 \end{pmatrix}, \quad C \times \begin{pmatrix} x_{world} \\ y_{world} \\ 1 \end{pmatrix} \equiv \begin{pmatrix} x_{cam} \\ y_{cam} \\ 1 \end{pmatrix} \tag{1}$$

$$C^{-1} \times T \times \begin{pmatrix} x_{proj} \\ y_{proj} \\ 1 \end{pmatrix} \equiv \begin{pmatrix} x_{world} \\ y_{world} \\ 1 \end{pmatrix}, \quad P = C^{-1} \times T \tag{2}$$

T is the projector-to-camera mapping using 4 corner points (x_{cam}, y_{cam}). Therefore the relationships between the three frames of reference corresponding to the projector, camera and projection screen.

$$P \times \begin{pmatrix} x_{proj} \\ y_{proj} \\ 1 \end{pmatrix} \equiv \begin{pmatrix} x_{world} \\ y_{world} \\ 1 \end{pmatrix}, \quad S \times \begin{pmatrix} x_{image} \\ y_{image} \\ 1 \end{pmatrix} \equiv \begin{pmatrix} x_{world} \\ y_{world} \\ 1 \end{pmatrix} \tag{3}$$

$$S^{-1} \times P \times \begin{pmatrix} x_{proj} \\ y_{proj} \\ 1 \end{pmatrix} \equiv \begin{pmatrix} x_{image} \\ y_{image} \\ 1 \end{pmatrix}, \quad \mathrm{W} = S^{-1} \times P \tag{4}$$

The pre-warped image is generated as warping function (*W*); W is defined as S, P. S is just a scaled rigid body transform since S maps a rectangle to another rectangle of the same aspect ratio. The keystone correction obtains the pre-warp, W by applying P^{-1} to the coordinates of the desired corrected image in the projected image frame; this is equivalent to applying $W = P^{-1} \times S$ to the original image.

2.4 Warping

Our system is warping a frame image of moving picture by DirectShow 9.0. DirectShow is Microsoft's newest and most exciting multimedia application builder [7]. Fig. 8 shows a graph for playing an AVI file. The AVI Splitter filter is splitting a frame of AVI file. We create a filter for warping a frame by a determinate warping function.

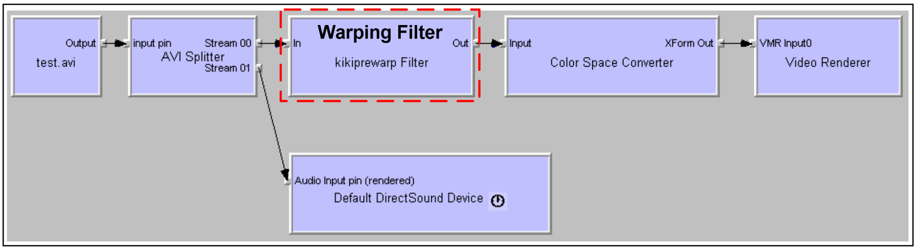


Fig. 8. Our filter graph for extracting and warping a frame image in DirectShow

The warping filter continuously takes 2 input values. One is a frame image, the other is warping function. This filter takes 2 steps (Fig. 9). The first step is warping an input image by the warping function (Fig. 9(b)). The second step: a pre-warp image is attached to points in 4 corners (Fig. 9(c)). Fig. 9(c) is a result image of warping filter.

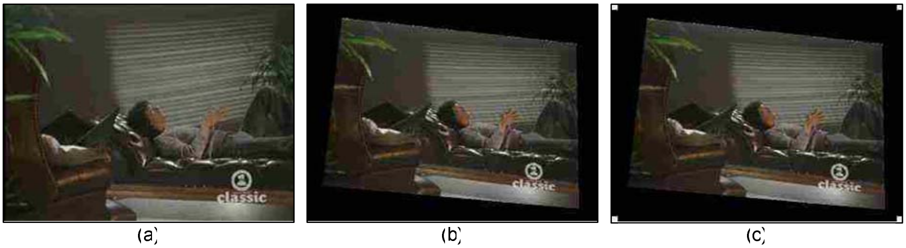


Fig. 9. Warping processing: (a) input image, (b) pre-warp image, and (c) add corner points to a pre-warp image

3 Results

We have built a system that includes an XGV (800x600) projector, a Logitech USB camera (640x480), and a Pentium 4 PC with ATI Radeon graphics board. Table 2 shows performance of the proposed system. The warping function consists of to detect 4 corner points and to compute a warping matrix. The corner point is detected at 0.1748-second intervals, and the warping matrix is computed at 0.1490-second intervals. Therefore, the warping function is computed at 0.3238-secont intervals, and a new pre-warp image is made at its intervals. Also, the warping process using warping function is required for a frame image at 0.0952-second intervals.

Table 2. Performance of the proposed system

	Warping Function		Image warping
Processing Time (s)	Corner points detection	Determining the warping function	0.0952 (10.5 fps)
	0.1748	0.1490	
	Total processing time		
	0.3238		

Fig. 9 shows result images of the real-time geometric calibration system. The proposed system corrects the distortion caused by oblique projection. As the projector is moving, the correcting image is continuously showed on plane surface (Fig. 9).

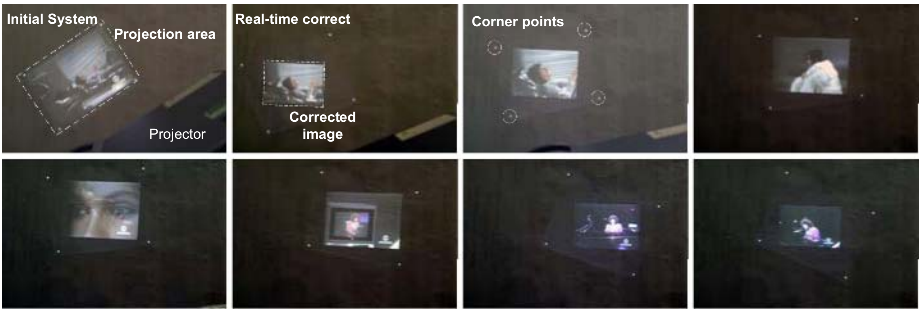


Fig. 9. Result image of the proposed system

4 Conclusions

The existing systems can not automatically calibrate in real-time during the motion of the projector, because warping function for real-time calibration requires the special o bject (e.g., maker, fiducials) on a projection surface and it is determined to manual by simply projecting a special pattern. However, our system provides a real-time automat ic calibration for oblique projection distortion. Our system projects both image and 4 corner points without the special object attached to the projection screen. This system takes 2 advantages. The first advantage is to provide a correcting image during the mo tion of the projector. The second advantage is automatically performing all process. T herefore, the proposed system is that it can be easily applied to ubiquitous computing.

Acknowledgments. This work was supported by grant No. (R01-2006-000-11214-0) from the Basic Research Program of the Korea Science & Engineering Foundation.

References

1. Weiser, M.: The Computer for the Twenty-First Century. Scientific American (1991) 94-100
2. D. Molyneaux and G. Kortuem.: Ubiquitous displays in dynamic environments: Issues and Opportunities. In Proceedings of Ubiquitous Display Environments, (2004)
3. K. Li, et al.: Early Experiences and Challenges in Building and Using A Scalable Display Wall System. IEEE Computer Graphics and Applications, vol. 20 (2000) 671-680
4. H. Chen, et al.: Scalable Alignment of Large-Format Multiprojector Displays Using Camera Homography Trees. In Proceedings of IEEE Visualization, IEEE CS Press, (2002) 339-346
5. G. Wallace, et al.: Tools and Applications for Large-Scale Display Walls. IEEE Computer Graphics and Applications, Vol. 25 (2005) 24-33
6. T.-J. Cham, et al.: Shadow Elimination and Occluder Light Suppression for Multi-Projector Displays. Computer Vision and Pattern Recognition (Demo), 2001
7. J. Summet et al.: Increasing the Usability of Virtual Rear Projection. In Proceedings of International Workshop on Projector-Camera Systems (PROCAMS), ICCV (2003)

8. Y. Ruigang and G. Welch.: Automatic Projector Display Surface Estimation Using Every-Day Imagery. In Proceedings of 9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, (2001)
9. C. Pinhanez.: The Everywhere Displays Projector:A Device to Create Ubiquitous Graphical Interfaces. In Proceedings of Ubiquitous Computing, Springer-Verlag, Berlin (2001) 315-331
10. G. Pingali et al.: User-Following Displays. In Proceedings of the IEEE International Conference on Multimedia and Expo, (2002)
11. Shree K. Nayar et al.: A Projection System with Radiometric Compensation for Screen Imperfections. In Proceedings of International Workshop on Projector-Camera Systems, (2003)
12. S. Borkowski, O. Riff, J. Crowley.: Projecting rectified images in an augmented Environment. In Proceedings of International Workshop on Projector-Camera Systems, (2003)
13. R. Raskar, et al.: iLamps: Geometrically Aware and Self Configuring Projectors. Presented at SIGGRAPH (2003)
14. Beardsley, P., et al.: Interaction Using a Handheld Projector. IEEE Computer Graphics and Applications, Vol. 25 (2005) 39-43
15. Sukthankar R., et al.: Smarter Presentations: Exploiting Homogra-phy in Camera-Projector Systems. In Proceedings of International Conference on Computer Vision, Vol. 1 (2001) 247-253
16. Michael L.: Programming Microsoft DirectShow. Wordware Publishing, (2002)

Face Occlusion Detection for Automated Teller Machine Surveillance

Daw-Tung Lin[1] and Ming-Ju Liu[2]

[1] Department of Computer Science and Information Engineering
National Taipei University
Sanshia, Taipei County, Taiwan
[2] Department of Computer Science and Information Engineering
Chun-Hwa University, Hsinchu, Taiwan

Abstract. Real time automatic alarm systems play an essential role in security management, as evidenced by the surveillance cameras installed in nearly all automated teller machines (ATMs). Whereas manual video surveillance requires constant staff monitoring, fatigue or distraction is a common human error. Therefore, this work presents an effective detection system for facial occlusion to assist security personnel in surveillance by providing both valuable information for further video indexing applications and important clues for investigating a crime. A series of methods that include identifying and segmenting moving objects is formed. The moving edge is then captured using change detection of the inter-frame difference and the Sobel operator. Next, a Straight Line Fitting (MSLF) algorithm is developed to merge the splitting blobs. Additionally, a mechanism involving moving forward or backward justification is used to determine whether an individual is approaching a camera. Moreover, the lower boundary of a head is computed, followed by use of an elliptical head tracker to match the head region. Finally, skin area ratio is calculated to determine whether the face is occluded or not. The proposed detection system can achieve 100% and 96.15% accuracy for non-occlusive and occlusive detection, respectively, at a speed of up to 20 frames per second.

1 Introduction

This paper focuses on a particular task, namely face occlusion detection for bank Automated Teller Machine (ATM) surveillance. Real time automatic alarm systems play an essential role in security management, as evidenced by the surveillance cameras installed in nearly all ATMs. In most bank robberies, the criminal with face occlusion is an obvious feature as shown in Fig. 1. An effective detection system for facial occlusion can be developed to assist security personnel in surveillance by providing both valuable information for further video indexing applications and important clues for investigating a crime. Our task is to develop a system which can detect such particular occasion of face occlusion and issue a warning message.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 641–651, 2006.

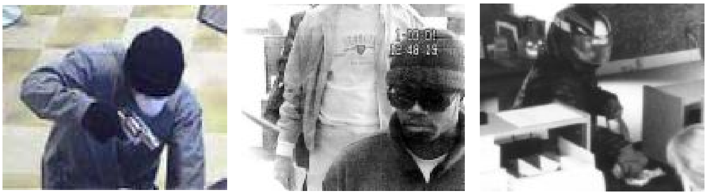


Fig. 1. Bank robbery snap shots from surveillance camera

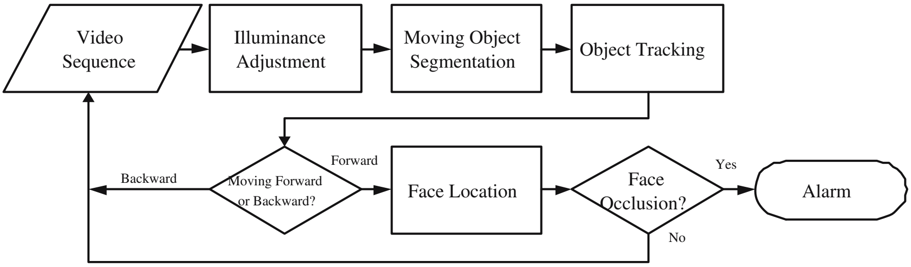


Fig. 2. Flow diagram of the proposed face occlusion detection system

To do so, a series of research subjects that include moving object segmentation, object tracking, and facial or non-facial feature detection must be investigated. Most of the moving object segmentation methods use background subtraction approaches based on either temporal or spatial information of the images [1,2,3]. Kim and Hwang [4] proposed a robust algorithm of video object planes (VOPs) by using a double-edge map. Ouyang et al. presented a neuro-fuzzy approach for segmentation of human objects in the image sequences [5]. The learned background model is used to extract foreground pixels. For example, Harwood et al. [2] developed a W4 system. Large number of tracking systems with various emphases have been reported by researchers. Pfinder [6] has been used to recover a 3-D description of person for tracking a single un-occluded person in a complex indoor environment. MIT's monitor system [7] uses an adaptive Gaussian mixture model to construct background scenes and mainly focuses on object classification, motion pattern learning, and abnormal activities detection. W4 system and its extension, HYDRA system [2], employs silhouette analysis to detect the body parts and track multiple people through occlusions. Face recognition is one of the most active research areas in pattern recognition in this decade. An extensive survey of still- and video-based face recognition approaches has been presented by Zhao et al. [8]. Most of the methods may be categorized into geometric feature-based, template-based, and machine learning-based techniques [9,10]. The problem of occluded face recognition has been studied by using partial feature representation and data reconstruction method [11,12].

In this paper, we develop a vision-based system to track people and detect face occlusion. The system flowchart is depicted in Fig. 2. In the next section, we describe how to identify and segment moving objects. We also introduce a Straight Line Fitting (MSLF) algorithm to merge the extracted splitting blobs. A mechanism involving tracking and moving forward/backward justification is derived to determine whether an individual is walking approaching a camera. Section 3 describes how we utilize the elliptical tracker to find the head region. In Section 4, we illustrate the scheme of skin area ratio calculation which is used to determine whether the face is occluded or not. The result of thorough simulations is shown in Section 5. Finally, conclusion remark is made in Section 6.

2 Moving Object Segmentation

2.1 Pre-process and Moving Object Edge Extraction

To reduce the amount of random noise caused by lighting change, we adopt the YUV color model and extract illuminance information [13]. Furthermore, the Y component is multiplied by 0.5 when converting the YUV space back to RGB color space. Next, the proposed motion edge extraction algorithm starts with getting the edge difference of the current frame (DE_n) by comparing two consecutive frames as follows.

$$DE_n(i,j) = \begin{cases} 0, \text{ if } |I_n(i,j) - I_{n-1}(i,j)| \leq T \\ \\ 1, \text{ otherwise} \end{cases} \tag{1}$$

where $I_n(i,j)$ denotes the intensity of pixel (i,j) in frame n, and T is a threshold. After the edge filtering operation, the result may contain edges of people, background, or noise. Thus, we proposed a procedure to get a more accurate motion edge(ME_n) by applying logic AND operation of the difference edge DE_n and the Sobel edge SE_n of the current frame by applying Sobel filtering [13]. Figure 3 illustrates the process of motion edge extraction step by step. As we can see, Fig. 3(e) becomes more accurate, which is meaningful for future usage.

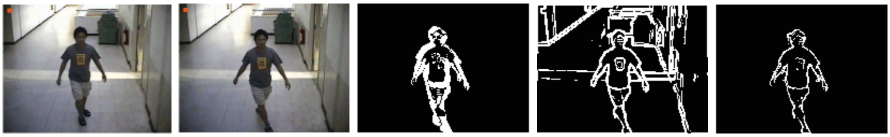

(a) (b) (c) (d) (e)

Fig. 3. Illustration of motion edge extraction process: (a) previous frame I_{n-1}, (b) current frame I_n, (c) difference edge DE_n, (d) Sobel edge SE_n of the current frame I_n, and (e) the motion edge ME_n obtained by logic AND operation of (c) and (d)

2.2 Motion Blob Extraction

To obtain correct region of the moving objects, we will search the intersection of horizontal and vertical candidates. The horizontal candidates (HC_n) are obtained by scanning the horizontal projection lines of the motion edges image

(ME_n). The vertical candidates(VC_n) will be found in the same manner by scanning the vertical lines. The concept of motion blobs extraction can be expressed as in Equation (2).

$$ME_n = (DE_n \cap SE_n), MB_n = (HC_n \cap VC_n) \tag{2}$$

We observed in many cases that the intersection blobs may be broken due to the projection of the defect edges. To get more complete blob area, we apply a 5×5 dilation and erosion morphology operator to the intersection area and continue to adopt the region growing algorithm iteratively [13]. Figure 4 depicts the process of the motion blobs (MB_n) extraction procedure. Figure 4(a) shows the motion edge of the current frame. Figure 4(b) and (c) present the horizontal and vertical candidates, respectively. Fig. 4(d) illustrates the intersection of Figs. 4(b) and (c). Figure 4(e) is the result of the morphology operation from Fig. 4(d).

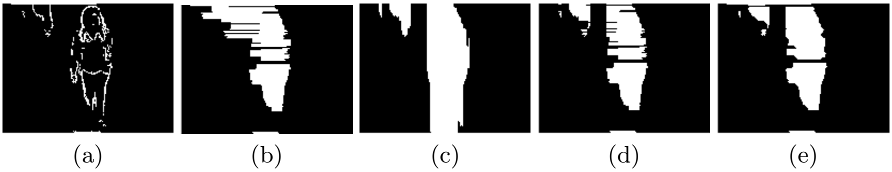


Fig. 4. The process of motion blobs extraction: (a) motion edge, (b) horizontal candidate, (c) vertical candidate, (d) intersection of (b)and(c), (e) result of the morphology operation from (d)

2.3 Merging by Straight Line Fitting

A flaw in the motion edge extraction process is that if the color of part of the background is same with the moving object, it may result in a broken motion edge. This will consequently split the moving object into more than one piece (see Fig. 5(c)). To overcome this drawback, we propose the following Merging by Straight Line Fitting (MSLF) method. A standard technique in mathematical and statistical modeling is to find a "least square" fitting to a set of data points in the plane. Suppose $\hat{x}$ is a least square solution to the system $Ax = b$ and $p = A\hat{x}$, then p is a vector in the vector space of A that is closest to b. Additionally, the

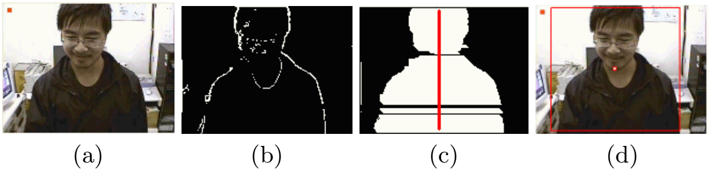


Fig. 5. The split blobs case: (a) original image, (b) motion edge, (c) its split blobs and straight line approximation, and (d) the final merged object

least squares problem $Ax = b$ will have a unique solution. Finding a straight line approximation is often called the method of least squares straight line fitting [14]. A useful functional approximation to the mapping $y_r = f(x_r)(0 \leq r \leq m)$ can be found, that is, we can find a function such as $y = ax^2 + bx + c$ such that y_r and $ax^2 + bx + c$ are equal or nearly equal for $0 \leq r \leq m$. By using the method of least squares straight line fitting described above, we assume that there exists a straight line approximation which is nearly vertical (see Figure 5 (c)). If the distance from the center of blob to the vertical line is less than a threshold, the blob will be merged. The merging process is described as follows. For the convenience of the follow up blob deletion and merge operation, we construct a link list ($BlobList_n$) to maintain the blobs in each frame. Each node of the link list contains the records of boundary, width, height and center coordinates of one blob. After the merging process, we will get more complete motion blobs. Figure 5(d) is the final result merged from three blobs.

Algorithm MSLF

```
for(i=0; i<$BlobList_n$ →Count; i++){
    blob1 = $BlobList_n$ →Items[i];
    The vertical line is y=a, a=blob1→center.x;
    for(j=i+1; j<$BlobList_n$ →Count; j++){
        blob2 = $BlobList_n$ →Items[j];
        if(|a - blob2→center.x| < blob1→width/3){
            merge blob2 into blob1;
            update blob information of blob1;
            delete blob2 from $BlobList_n$;
        }
    }
}
```

2.4 Moving Forward/Backward Identification

The final goal of the proposed system is to detect the facial occlusion of the people for ATM. We are only interested in detecting people who are facing the camera in the surveillance scene. To determine whether a person is walking forward to the camera, we proposed a forward/backward identification criteria.

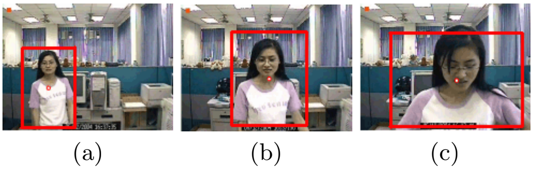

(a) (b) (c)

Fig. 6. Object moves toward camera. The ratio of width to height of (a), (b) and (c) are 0.6234, 0.8068 and 1.4138, respectively.

Let $blob_i width_n$ and $blob_i height_n$ be the width and height of blob i in the nth frame, respectively. The ratio of width to height is expressed as $blob_i ratio_n = blob_i width_n / blob_i height_n$. We define the ratio difference RD of consecutive frames $n-1$ and n as $RD_n^i = blob_i ratio_n - blob_i ratio_{n-1}$. The changes of the ratio difference are shown in Fig. 6. When the ratio difference of the same object becomes larger, it implies that the object is moving forward to the camera. To tolerate the measurement error of center coordinates and ratio, we define a tolerance threshold parameter $FwdBwd$. We set $FwdBwd$ to be 5 through experimental statistics. By judging the positive or negative sign of the parameter $FwdBwd$, we can identify whether the person is moving forward or backward to the camera. If $FwdBwd_n^i > 0$, we can conclude that the person is moving forward to the camera. Otherwise, the person is moving backward. This is a very important clue, because the system only needs to find the face of a person who is moving forward to the camera. The criteria are listed in below.

```
if ($RD_n^i$ >0) and ($FwdBwd_n^i$ <5))
    $FwdBwd_n^i$ = $FwdBwd_n^i$+1;
if ($RD_n^i$ <0) and ($FwdBwd_n^i$ >-5))
    $FwdBwd_n^i$ = $FwdBwd_n^i$-1;
```

3 Face Location Confirmation

In this research, we have designed three verification steps to confirm the face target when the face of a person is initially located. First, we compute the condition of the blob position and area ratio to verify if it is a face candidate. Secondly, we find the lower boundary of the head to restrict the searching window. Finally, we use the elliptical head tracker to determine the face region. After the face location is confirmed, we adopt the face center coordinates information to track that face in the follow-up frames.

Generally, the head contours can be divided into two categories: shape with obvious neck and smooth curve with hairs covering the neck. We need to find short horizontal projection lines which are close to the shoulder by searching from top to bottom in the defined object region. In the short hair case, these candidates must all fall in the valley of contours. We search the range from $0.125 * h$ to $0.75 * h$ and apply elliptical approximate algorithm [15] to track the head. In our experiments, we used an elliptic head model $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ with a fixed aspect ratio of 1.6, where (x,y) means the center coordinate, a and b denote one half of the minor and major diameters, respectively. The goodness of the elliptical match depends upon the gradient magnitude around the perimeter of the ellipse. In this paper, we use the motion edge ME_n described in Section 2.1 to match with the perimeter of ellipse. The motion edge is expressed by an array of 1's and 0's representing the blob edge pixel and the background pixel, respectively.

$$\begin{aligned} s^* = arg\ max_{s \in S}\{\tfrac{1}{4b}\textstyle\sum_{j=1}^{b}(ME_n(x+i,y+j) + ME_n(x-i,y+j) \\ +ME_n(x+i,y-j) + ME_n(x-i,y-j), \end{aligned} \quad (3)$$

where $i = \frac{a\sqrt{b^2-j^2}}{b}$. The search space S is the set of all states located within the search window. When a face is initially matched, S can be described as:

$$S = \{s : (x - a > b_{left})\&(x + a < b_{right}), (y - b > b_{top})\&(y + b < head_{lb})\} \tag{4}$$

where $2/3\sigma \leq a \leq \sigma$,$2/3 * 1.6\sigma \leq b \leq 1.6\sigma$. b_{left}, b_{right},b_{top}, and head$_{lb}$ denote the left, right, top, and lower boundary of a blob, respectively. σ is estimated by the upper horizontal projection of the body: $\sigma = \frac{1}{2}(\frac{1}{head_{lb}-b_{top}} \sum_{j=b_{top}}^{head_{lb}} P_h[j])$, where $P_h[j]$ denotes the horizontal projection.

4 Face Occlusion Detection

Based on the best matched elliptical area of head, the face region is defined as in Fig. 7. Point $C(x, y)$ is the center of the ellipse. The ellipse represents the head region, points P$_1$, P$_2$, P$_3$, and P$_4$ indicate the four corners of the face region. The y coordinate of P$_3$ and P$_4$ is obtained as $y + \frac{b\sqrt{a^2-(x-a)^2}}{a}$. We define the ratio of skin color area to face region as in Equation (5) to determine whether the face is occluded or not. A face is determined as non-occluded if the skin area ratio ≥ 0.6. Otherwise, the face is occluded. The threshold 0.6 is chosen through extensive experiments.

$$Skin\_ratio = \frac{\text{area of skin color in face region}}{\text{face area}} \tag{5}$$

For skin color identification, both normalized RGB color model (rgb) and RGB color model are used. We adopt the normalized RGB range mentioned in [16] to discriminate the skin color: $0.36 \leq r \leq 0.465$ and $0.28 \leq g \leq 0.363$. The range described above contains black and very dark brown color. To further exclude out those dim colors, we define additional rules: $\frac{R+G+B}{3} < 28$ and $B > 28$ in original RGB color model. The former constrain is to exclude the black pixels. The later condition restricts the blue color range, it aims at removing the very dark brown color.

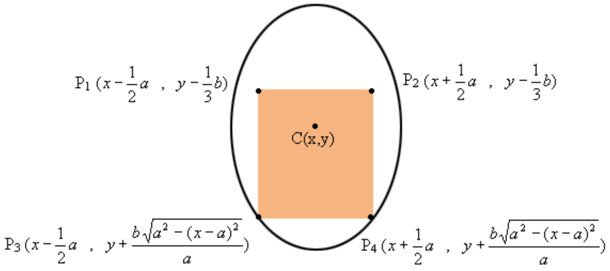

$P_1\ (x-\frac{1}{2}a\ ,\ y-\frac{1}{3}b)$
$P_2\ (x+\frac{1}{2}a\ ,\ y-\frac{1}{3}b)$
C(x,y)
$P_3\ (x-\frac{1}{2}a\ ,\ y+\frac{b\sqrt{a^2-(x-a)^2}}{a})$
$P_4\ (x+\frac{1}{2}a\ ,\ y+\frac{b\sqrt{a^2-(x-a)^2}}{a})$

Fig. 7. The face region of the best matched elliptical area. Skin color region indicates the face region.

5 Experimental Results

The proposed facial occlusion detection system is implemented on a low-end PC (Pentium III 1.0 GHz CPU). A warning alarm will be issued once the occlusion case is detected. We used a DVR to capture video sequences with setting of bank ATM. For the sake of real time performance, the video image size sequence is down-scaled by 1/4. The result is a 160x120 image per frame. The developed detection system works at the speed of 5-20 frames per second.

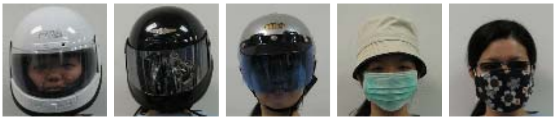


Fig. 8. Five examples of face occlusion

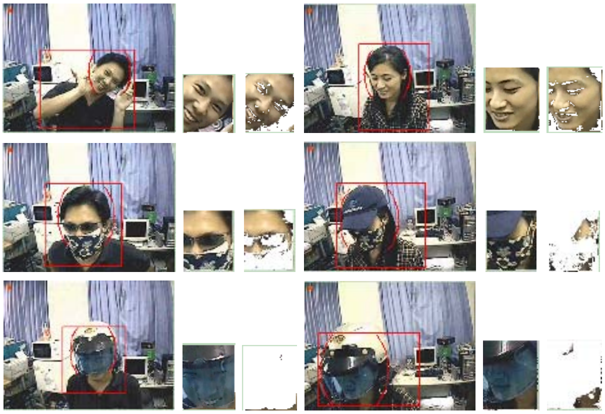


Fig. 9. Six examples of face occlusion detection results

We have tested on 173 video clips including 130 occlusive cases and 43 non-occlusive cases. The occlusive case video sequences were recorded from ten people (five males and five females), each with thirteen face occlusion scenarios. Each person walks either forward or backward to the ATM and pretend doing transaction or cashing. In addition, 33 people were tested for non-facial occlusion. Figure 9 presents the results of six facial occlusion detection scenarios. The red circle delineates the detected head area. The square windows show the face area and skin color area accordingly. Table 1 shows the performance of detection. The proposed system achieves 95.35% accuracy for non-occlusive detection and 96.15% accuracy in occlusive detection, the overall performance is 95.95%. We also compared the detection performance with other methods [17] and [18] as

Table 1. Face occlusion detection hit ratio

Face type	Sequence number	correct	Hit ratio
Non-occlusive	43	41	95.35%
Occlusive	130	125	96.15%
Overall	173	166	95.95%

Table 2. Performance comparison of the proposed system with other methods

Method	occlusive cases	non-occlusive cases
Lin and Lee[17]	96.2% 257/267 frames	97% 132/136 frames
Lin and Fan[18]	100% 1/1 video clips	100% 3/3 video clips
Our Method	95.35% 125/130 video clips	96.15% 41/43 video clips

shown in Table 2. Lin and Lee compute the gray values of eyes and mouth area for justifying their appearance [17]. Their overall accuracy rate was 96.5% out of 403 test images. Lin and Fan used the skin color distribution ratio to determine if the face is occluded [18]. The accuracy rate was 100% (4/4 video clips). As we can see from Table 2, our system performance is comparative to the others. While, our simulation result is more objective with extensive amount of test sequences and more realistic environments.

6 Conclusion

We have implemented a real time face occlusion detection system. This work presents an effective detection system for facial occlusion to assist security personnel in surveillance by providing both valuable information for further video indexing applications and important clues for investigating a crime. First, static and dynamic edges are combined to detect moving objects. Then a standard least squares straight line fitting method is used to merge motion blobs. Moving forward or backward is determined based on the aspect ratio of bounding box. An ellipse model is fitting to the candidate face by maxmizing the gradient magnitude around the perimeter. Finally, face occlusion is performed based on the ratio of skin color in the face where the skin color is determined by a simple thresholding in R-G color space. The proposed detection system can achieve an overall accuracy of 96.43% at a speed of 5-20 frames per second. The proposed facial occlusion detection system can be applied for surveillance of ATMs with a complex background.

Acknowledgement

This work is supported in part by Ministry of Economic Affairs, Taiwan, under grant no. 95-EC-17-A-02-S1-032, and by National Science Council. Taiwan, under grant no. NSC 94-2622-E-305-001-CC3.

References

1. Elena Stringa and Carlo S. Regazzoni. Real-time video-shot detection for scene surveillance applications. *IEEE Transactions on Image Processing*, 9(1):69–79, January 2000.
2. Ismail Haritaoglu, David Harwood, and Larry S. Davis. W4:real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, August 2000.
3. Osama Masoud and Nikolaos P. Papanikolopoulos. A novel method for tracking and counting pedestrians in real-time using a single camera. *IEEE Transactions on Vehicular Technology*, 50(5):1667–1278, September 2001.
4. Changick Kim and Jenq-Neng Hwang. Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Transactions on Circuits And Systems for Video Technology*, 12(2):122–129, February 2002.
5. Shie-Jue Lee, Chen-Sen Ouyang, and Shih-Huai Du. A neuro-fuzzy approach for segmentation of human objects in image sequences. *IEEE Transactions on Systems, Man, and CybernaticsXPart B*, 33(3):420–437, June 2003.
6. C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intrelligence*, pages 780–785, 1997.
7. C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, pages 747–757, 2000.
8. W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, December 2003.
9. R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1042–1052, 1993.
10. M. Fukumi, Y. Mitsukura, and N. Akamatsu. A design of face detection system by lip detection neural network and skin distinction neural network. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 2789–2793, October 2000.
11. T. Kurita, M. Pic, and T. Takahashi. Recognition and detection of occluded faces by a neural network classifier with recursive data reconstruction. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 53–58, 2003.
12. I.B. Ciocoiu. Occluded face recognition using parts-based representation methods. In *Proceedings of the 2005 European Conference on Circuit Theory and Design*, volume 1, pages 315–318, 2005.
13. R. C. Gonzalea and R. E. Woods. *Digital Image Processing, 3nd.* Addision-Wesley, 2003.
14. Anton and Rorres. *Elementary Linear Algebra, Application Version, 8th Edition.* Wiley, 2002.

15. Stan Birchfield. Elliptical head tracking using intensity gradients and color histogram. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–237, June 1998.
16. Yanjiang Wang and Baozong Yuan. A novel approach for human face detection from color images under complex background. *The Journal of the Pattern Recognition Society*, 34:1983–1992, 2001.
17. Hsin-Chien Lin. Detection of faces with covers in a vision-based security. Master's thesis, National Chiao-Tung University, Taiwan, June 2003.
18. Huang-Shan Lin. Detection of facial occlusions by skin-color based and local-minimum based feature extractor. Master's thesis, National Central University, Taiwan, June 2003.

Automatic Pose-Normalized 3D Face Modeling and Recognition Systems

Sunjin Yu[1], Kwontaeg Choi[2], and Sangyoun Lee[1]

Biometrics Engineering Research Center,
[1] Dept. of Electrical and Electronics Engineering,
[2] Dept. of Computer Science, Yonsei University
134 Shinchon-dong, Seodaemun-gu, Seoul 120-749, South Korea
{biometrics, choikt, syleee}@yonsei.ac.kr

Abstract. Pose-variation factors present a big problem in 2D face recognition. To solve this problem, we designed a 3D face acquisition system which was able to generate multi-view images. However, this created another pose-estimation problem in terms of normalizing the 3D face data. This paper presents an automatic pose-normalized 3D face data acquisition method that is able to perform both 3D face modeling and 3D face pose-normalization at once. The proposed method uses 2D information with the AAM (Active Appearance Model) and 3D information with a 3D normal vector. The proposed system is based on stereo vision and a structured light system which consists of 2 cameras and 1 projector. In orsder to verify the performance of the proposed method, we designed an experiment for 2.5D face recognition. Experimental results showed that the proposed method is robust against pose variation.

1 Introduction

Although current 2D face recognition systems have reached a certain level of maturity, performance has been limited by external conditions such as head pose and lighting conditions. To alleviate these factors, 3D face recognition methods have recently received significant attention, and the appropriate 3D sensing techniques have been highlighted [1][2].

Many 3D data acquisition methods that use general-purpose cameras and computer vision techniques have been developed [3][4]. Since these methods use general-purpose cameras, they may be effective for low-cost 3D input sensors that can be used in 3D face recognition systems. Previous approaches in the field of 3D shape reconstruction in computer vision can be broadly classified into two categories; active and passive sensing.

Although the stereo camera, a kind of passive sensing device, infers 3D information from multiple images, the human face contains an unlimited number of features. Because of this, it is difficult to use dense reconstruction with human faces. There is also a degree of ambiguity when matching point features. This is known as the matching problem [5]. Therefore, passive sensing is not an adequate choice for 3D face data acquisition.

Active sensing, however, uses a CCD camera to project a special pattern onto the subject and reconstruct shapes by using reflected pattern imaging [6]. Because active

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 652 – 661, 2006.

sensing is better at matching ambiguity and also provides dense feature points, it can act as an appropriate 3D face-sensing technique.

However, in general, a 3D reconstruction device requires pose-normalization in order to recognize objects after acquisition. Some 3D recognition systems assume that a normalized 3D database exists already. They may also carry out another pose-normalization step after the initial acquisition step [7][8][9][10][11]. This represents a 3D head pose estimation problem in 3D face recognition systems. Solutions to this kind of problem can be roughly divided into feature-based and appearance-based methods. Feature-based methods try to estimate the position of significant facial features such as the eyes, the nose or the mouth in the facial range or intensity image. Appearance-based methods consider the facial image as a global entity [12].

This paper presents a new method for automatic pose-normalized 3D data acquisition which is able to perform both 3D face data acquisition and pose-normalization at once. In the proposed system, both binary and color time-multiplexing patterns were used. Once the correspondence problem was solved, the 3D range data was computed by using triangulation. Triangulation is a well-established technique for acquiring range data with corresponding point information [3]. For pose-normalization, the pose-estimation method was used with the AAM (Active Appearance Method) [13].

This paper is organized as follows. Section 2 presents background information, and the 3D face modeling step is discussed in Section 3. In Section 4, the automatic pose-normalized 3D face data acquisition method is proposed and Section 5 comprises the experimental results. Finally, Section 6 concludes the paper.

2 Background

2.1 Camera Calibration

Calibration refers to the process of estimating the parameters that determine a projective transformation from the 3D space of the world onto the 2D space of an image plane. A set of 3D-2D point pairs for calibration can be obtained with a calibration rig. If at least six point pairs are known, the calibration matrix can be uniquely determined. However, in many cases, due to errors, it is better to use far more than six point pairs. Although this may result in what is known as the over-determined problem, we used 96 point pairs to ensure high accuracy in our experiments. Finally, the stereo camera system was calibrated with the DLT (Direct Linear Transform) algorithm [3][4].

2.2 Epipolar Geometry

Epipolar geometry refers to the intrinsic projective geometry between two given views. It is independent of scene structure, and depends only on the camera's internal parameters and relative pose [4]. The fundamental matrix F encapsulates this intrinsic geometry. If a point in 3 space X is imaged as x in the first view, and x' in the second, then the image points can be said to satisfy the relation $x'^T Fx = 0$, $l' = Fx$ [4].

The fundamental matrix is also an algebraic representation of epipolar geometry. It derives the fundamental matrix from the mapping between a point and its epipolar line, and then specifies the properties of the matrix [4]. Using the normalized 8-point algorithm [4] [14], we extracted the fundamental matrix.

2.3 Structured Light System

Stereo vision is based on capturing a given scene from two or more points of view and then finding the correspondences between the different images in order to triangulate the 3D position. However, difficulties in finding the correspondences may arise, even when taking into account epipolar constraints. In addition, active stereo vision using light can be divided into time-multiplexing, spatial neighborhood and direct coding [6]. It is possible to use both binary and color time-multiplexing coded patterns to solve the correspondence problem. The idea of these methods is to provide distinguishing features in order to find the correspondence pairs.

2.4 Active Appearance Model

Each shape sample x can be expressed by means of a set of shape coefficients as shown in equation (1) and texture g as shown in equation (2).

$$x = \overline{x} + P_s b_s \tag{1}$$

$$g = \overline{g} + P_g b_g \, , \tag{2}$$

where $\overline{x}$ is the mean shape, P_s is the matrix with eigenvectors on the shapes, b_s is a shape coefficient, $\overline{g}$ is the mean texture, P_g is the matrix with eigenvectors on the texture, and b_g is a texture coefficient. Since the shapes and gray-level textures can be correlated with each other, the shape and texture coefficients can be concatenated into a vector b as shown in equation (3).

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = Qc \tag{3}$$

In equation (3), Q represents the eigenvectors and c is a vector of the appearance parameters which control both the shape and gray-levels. Thus all images are modeled with a vector c which describes both the shape and gray-level texture variations together.

With the given face image, it was necessary to search for a 2D face model that minimized the texture error [13]. After AAM matching, b_s , b_g and c were extracted and used as shape coefficients, $\overline{g}$ represented the mean texture, and P_g was the matrix with eigenvectors on the texture. All of these were used as features to identify faces. To improve recognition performance, we used a feature extraction technique which combined a generic AAM with a person-specific AAM. In training, one generic model was constructed which captured the variation of all the training faces, and a specific model was constructed for each individual to model the variation of only one person.

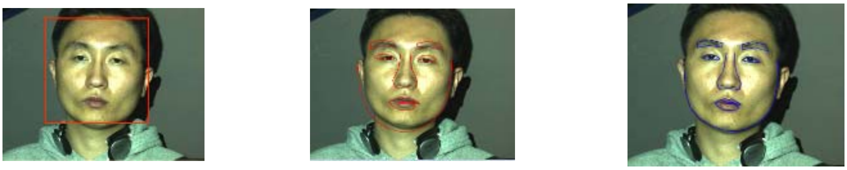
Generally the AAM is able to provide information on the principal axes of the training data since the main idea of the AAM is PCA. The axes extracted from the generic model are efficient ways to identify faces; meanwhile the axes extracted from the person-specific models are efficient ways to track the faces of the corresponding person. Thus, the person-specific model is more suitable for tracking faces whereas the generic model is more suitable for extracting facial features for recognition.

With the given face image, we first used the person-specific model to track and extract the shape aligned to that face. In the first frame we tried all person-specific models and then selected one model by using the minimum matching error. We applied a generic AAM model to the selected face region in order to extract the shape and texture features for face recognition. Fig. 1 shows the way to get the geometric information of a face by using the AAM. To find the initial shape position, we used the face detection method proposed by the Viola-Jones cascade detector [15].

3 3D Face Modeling

3.1 3D Face Data Acquisition System

The simplest active technique is to use a single-line stripe pattern, which greatly simplifies the matching process, even though only a single line of 3D data points can be obtained with each image shot. To speed up the acquisition of 3D range data, it is necessary to adopt a multiple-line stripe pattern instead. However, the matching process then becomes much more difficult. One other possibility is to use color information to simplify this difficulty [6].



(a) Face detection (b) Initial shape (c) Shape fitting

Fig. 1. Face geometric information using the adaboost algorithm and the AAM

Furthermore, when using the single-camera approach, it is necessary to find the correspondence between the color stripes projected by the light source and the color stripes observed in the image. In general, due to the different reflection properties (or surface albedos) of varying object surfaces, the color of the stripes recorded by the camera is usually different from that of the stripes projected by the light source (even when the objects are perfectly Lambertian)[16]. It is difficult to solve these problems in many practical applications.

However, this does not affect our system (which consists of two cameras and one projector) if the object is Lambertian, because the color observed by the two cameras will be the same, even though this observed color may not be exactly the same as the color projected by the light source. Therefore, by adding one more camera, the more difficult problem of lighting-to-image correspondence is replaced by an easier problem of image-to-image stereo correspondence. Here, the stereo correspondence problem is also easier to solve than traditional stereo correspondence problems because an effective color pattern has been projected onto the object.

3.2 Pattern Design

The binary sequential pattern uses two illumination levels which are coded as 0 and 1 and the color sequential pattern uses color instead of these two illumination levels [6]. Previous research has shown that the color model is an effective model for stereo matching [17][18][19]. We adopted the HSI model for color-coded sequential pattern generation. Using the line features and the HSI color model, a set of unique color encoded vertical stripes was generated. Each color-coded stripe was obtained as follows. The stripe color was denoted as shown in equation (4).

$$stripe(\rho,\theta) = \rho e^{j\theta}, \tag{4}$$

where ρ is the saturation value and θ is the hue value of the HS polar coordinate system. We used only one saturation value (saturation=1) because there was enough hue information to distinguish each stripe for the matching process. Finally, the stripe color equation was defined by using equation (5).

$$color(m,n) = e^{j(mH_{jmp}+\varepsilon n)}, \tag{5}$$

where m represents the set element, n is the set number, and ε is the hue perturbation. Next, the color-coded sequence was obtained as follows. We determined that the hue distance was 144°. The next set elements were then used sequentially.

In this paper, we generated codes in the face region. We used seven images for the binary pattern and three images for the color pattern. The binary pattern generated $2^7 = 128$ codes and the color pattern generated $5^3 = 125$ codes.

3.3 Absolute Code Interpolation

After triangulation, we obtained the 3D reconstructed face data. However, this data was sparse, because we had to deal with a thinned code in each view. Thinning refers to the process of reducing the ambiguity of the corresponding problem. If we did not use the thinning step, the 1-to-many matching problem would have been encountered. Fig. 2 shows the reconstructed sparse 3D face results. However, in order to recognize the images, dense reconstruction is required. Therefore, we used absolute code interpolation to discover which human's face surface comprised the continuous region and when the neighbor region's depth was similar. The thinned codes in each view were interpolated by linear interpolation. Fig. 3 shows an example of code interpolation. Fig. 4 shows an example of reconstructed dense 3D face data using the proposed absolute code interpolation. The reconstructed 3D face data consists of a 3D point cloud and a texture map.

4 Automatic Pose-Normalized 3D Face Modeling

4.1 3D Face Pose-Normalization

The proposed automatic normalized 3D face data acquisition method uses 2D and 3D information fusion. First, we found the eye region, nose region and mouth region in the 2D image by using the AAM. Second, we reconstructed the 3D face. Third, because we knew the 2D and 3D relation equations, we were able to find the center of the 3D eyes, the top of the nose and the center point of the mouth. Fourth, using the 3D eye and nose top points, we produced a mesh plane. Fifth, using this mesh plane, we were able to find the normal vector.

Finally, the problem of DOF 3 against translation was solved by nose tip rearrangement and the problem of DOF 3 against rotation was solved by normal vector rearrangement. By obtaining the three points $\mathbf{P}_0$, $\mathbf{P}_1$ *and* $\mathbf{P}_2$ (two eyes' mid points and lip's mid point), we were able to make the 3D normal vector as follows [20]:

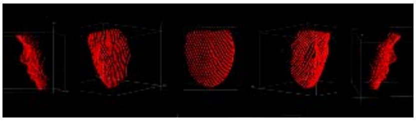
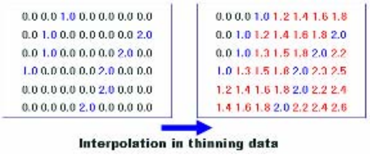


Fig. 2. Reconstruction of 3D face



0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	2.0
0.0	1.0	0.0	0.0	0.0	2.0	0.0
1.0	0.0	0.0	0.0	2.0	0.0	0.0
0.0	0.0	0.0	0.0	2.0	0.0	0.0
0.0	0.0	0.0	2.0	0.0	0.0	0.0

0.0	0.0	1.0	1.2	1.4	1.6	1.8
0.0	1.0	1.2	1.4	1.6	1.8	2.0
0.0	1.0	1.3	1.5	1.8	2.0	2.2
1.0	1.3	1.5	1.8	2.0	2.3	2.5
1.2	1.4	1.6	1.8	2.0	2.2	2.4
1.4	1.6	1.8	2.0	2.2	2.4	2.6

Fig. 3. Code Interpolation

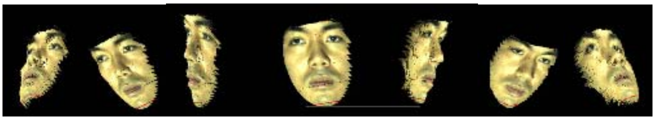


Fig. 4. Reconstructed dense 3D face data using the proposed absolute code interpolation

$$\mathbf{N} = \frac{(\mathbf{P}_1 - \mathbf{P}_0)\times(\mathbf{P}_2 - \mathbf{P}_0)}{\|(\mathbf{P}_1 - \mathbf{P}_0)\times(\mathbf{P}_2 - \mathbf{P}_0)\|}, \tag{6}$$

where N is the 3D normal vector. The process of the proposed method is as follows:

1. Find the 2D geometry features such as eye, nose and mouth region in the 2D image.
2. Reconstruct the 3D face data.

3. Find the 3D geometry features in the 3D space using the relation between the 2D geometry features and the 3D position.
4. Search the nose tip under the nose region using the 2D nose information.
5. Move the 3D face data's center to the nose tip for translation.
6. Check the eye position and align the 3D face data for the Z axis horizontally.
7. Calculate the 3D normal vector using the two eye region's mid points and the lip region's mid point.
8. Rotate the 3D normal vector with the 3D face data for the X and Y axis.

4.2 Procedure

3D face modeling can be divided into an off-line and an on-line process. Calculating the camera matrix and the fundamental matrix is an off-line process. Because these are calculated before the on-line process, they are not related to running-time. After the off-line process, it is possible to carry out the on-line process, which is related to running-time. The on-line process can be roughly divided into 2D processing and 3D processing. 2D processing consists of capturing the image pairs, detecting the face region, extracting the 2D face geometry features, thinning, absolute code interpolation and finding corresponding pairs. 3D processing consists of triangulation as well as the automatic pose-normalized face modeling step. The procedure is as follows:

1. Choose the pattern type (binary or color).
2. Project the coded patterns onto the human face and capture the stereo reflected pattern images.
3. Detect the face region using the adaboost algorithm in order to provide the initial shape position.
4. Extract the 2D facial geometry features such as eyes, nose, and lip regions using the initial shape position and the AAM.
5. Generate the code according to the pattern type (binary or color).
6. Thin the generated code in order to reduce the correspondence problem.
7. Interpolate the thinned code.
8. Find the corresponding pairs using the fundamental matrix.
9. Triangulate using the camera matrix.
10. Find the 3D normal vector and the nose tip using the 2D facial geometry information.
11. Normalize the pose using the 3D normal vector.

Items 1~8 refer to 2D processing steps and items 9-11 refer to 3D processing steps.

5 Experimental Results

5.1 3D Face Database

We created a 3D face database of fifteen persons. Each person was captured at five different pose variations: front, up, down, left and right. Also, each person was subjected to binary sequential and color sequential reconstruction. The color sequential reconstructed data used a train and a gallery, while the binary sequential reconstructed data used a query. After normalization, we obtained the normalized 3D face data. Fig. 5 shows the geometric information and the 3D normal vector in the 3D space. Fig. 6 shows an example of the 3D face database, both with and without the proposed normalized method.

5.2 Recognition Results

To verify the proposed pose-normalized method, we compared two types of recognition experiments. The first experiment was done in order to recognize the reconstructed 3D face data without pose-normalization. The second experiment was done to recognize the reconstructed 3D face data with pose-normalization. Both experiments projected the 3D to 2D space with texture. This method can apply the existing 2D face recognition algorithm easily. It is a 2.5D face recognition method. The gallery included reconstructed 3D face data using a color sequential pattern and the query included reconstructed 3D face data using a binary sequential pattern. Table 1 shows the used sets. To recognize them, we used the AAM shape information and Euclidean distance. Table 2 shows the identification results using the AAM shape information. After normalization, the recognition rate improved by about 35%.

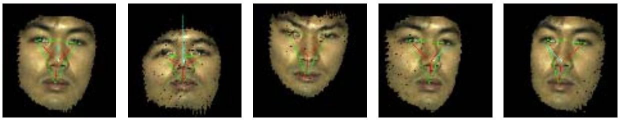
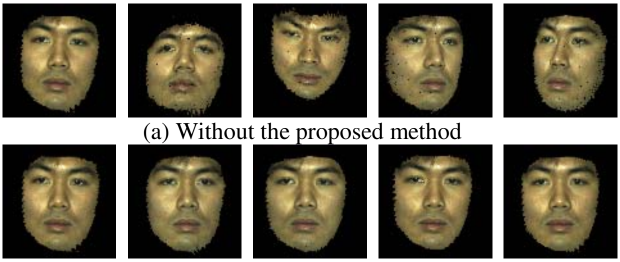


Fig. 5. Geometric information and the normal vector in the 3D face space



(a) Without the proposed method

(b) With the proposed method

Fig. 6. Reconstructed 3D face database without and with pose-normalization

Table 1. Used sets and types

Used set	Type	Number of data items
Color sequential	Gallery	5×15=75
Binary sequential	Query	5×15=75

Table 2. Identification results

Set	Normalization	Number of data items
Projected 2D texture	OFF	60 % (45/75)
Projected 2D texture	ON	94.67 % (71/75)

6 Conclusions

This paper presents an automatic pose-normalized 3D face data acquisition method and a 2.5D face recognition system. We assumed that the user's pose variation range was $\pm 35^{\circ}$ in each axis and identification system was trained for only resisted people using AAM.

We produced a full face recognition system from 3D data acquisition to recognition so that we produced 3D modeling which used a pose-normalized method. Processing time is about ± 10 seconds which are depended matching points.

Our system consisted of two cameras and one projector, with expansion capabilities. Because the projector provided the features, we were easily able to change features such as lasers, infra-red systems and so on.

To normalize the pose factor, we used the 2D and 3D information. This method showed some advantages. Both 2D and 3D information can present both advantages and disadvantages. Therefore, we used the 2D and 3D information to provide advantages and to offset disadvantages. We also used a 3D point cloud and texture map. The recognition method was 2.5D face recognition, which is very useful because 2D recognition has lower computational cost than 3D recognition methods and there are powerful recognition methods in the 2D space.

The proposed pose-normalization method shows robust results against pose variation. Experimental results show that the normalized recognition rate improved by about 35% when compared with un-normalized recognition.

In future work, we will consider other recognition methods in the 2D space such as the SVM (Support Vector Machine), ICA (Independent Component Analysis), LFA (Local Feature Analysis) and so on. Finally, we hope to design a full 3D face data recognition and 2D and 3D multimodal recognition system.

Acknowledgements. This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei University, Seoul, South Korea.

References

1. H.S. Yang, K.L. Boyer and A.C. Kak, "Range data extraction and interpretation by structured light," Proc. 1st IEEE Conference on Artificial Intelligence Applications, Denver, CO, pp. 199–205, 1984.
2. K.L. Boyer and AC. Kak, "Color-encoded structured light for rapid active ranging," IEEE Trans. Pattern Analysis and Machine Intelligence, pp.14–28, 1987.
3. Emanuele Trucco and Alessandro Verri, "Introductory Techniques for 3-D Computer Vision," Prentice Hall, 1998.

4. R. Hartley and A. Zisserman, “Multiple view Geometry in computer vision,” Cambridge University Press, 2000.
5. D. Scharstein and R. Szeliski, “Stereo Matching with Nonlinear Diffusion”, International Journal of Computer Vision, 28(2), pp. 155 – 174, 1998.
6. J. Salvi, J. Pags and J. Batlle, “Pattern Codification Strategies in Structured Light Systems,” Pattern Recognition 37(4), pp. 827–849, April 2004.
7. S. Malassiotis and M. G. Strintzis, “Pose and illumination compensation for 3D face recognition,” Proc. Int. Conf. Image Processing, 1, pp. 91–94, 2004.
8. S. Malassiotis and M. G. Strintzis, “Robust Face Recognition Using 2D and 3D Data: Pose and Illumination Compensation,” Pattern Recognition, Jan. 2005.
9. Chang, K.I., Bowyer, K.W. and Flynn, P.J. “Multi-Modal 2D and 3D Biometrics for Face Recognition,” Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop, pp. 187-194, Oct. 2003.
10. C. Xu, Y. Wang, T.Tan and L.Quan, “A New Attempt to Face Recognition Using Eigenfaces”, Proc. ACCV, pp.884~889, 2004.
11. H. Song, S. Lee, J. Kim and K.Sohn, “3D sensor based face recognition,” Applied Optics, Vol. 44, No. 5, pp. 677-687, Feb. 2005.
12. H. Song, “3D Head Pose Estimation and Face Recognition Using Range Image”, Ph.D Thesis, Yonsei University, South Korea, 2005.
13. T. F. Cootes, G. Edwards and C. Taylor, “Active Appearance Models,” IEEE Trans. On Pattern Analysis and Machine Intelligence, 23(6), pp. 681–685, 2001.
14. Hartley, R., "In Defence of the 8-points Algorithm", in Proceedings Fifth International Conference on Computer Vision, Cambridge, Mass., June 1995.
15. P. Viola and M. Jones, “Robust real-time face detection,” International Journal of Computer Vision 57(2), pp. 137–154, 2004.
16. D. Shin and J. Kim, “Point to Point Calibration Method of Structured Light for Facial Data Reconstruction”, First International Conference, ICBA 2004, pp. 200-206, Jul. 2004.
17. C.H. Hsieh, C.J. Tsai, Y.P. Hung and SC. Hsu, “Use of chromatic information in region-based stereo,” Proc. IPPR Conference on Computer Vision, Graphics, and Image Processing, Nantou, Taiwan, pp. 236–243, 1993.
18. R.C. Gonzales and R.E. Woods, “Digital Image Processing,” Addison-Wesley, Reading, MA, 1992.
19. C. Chen, Y. Hung, C. Chiang and J. Wu, “Range data acquisition using color structured lighting and stereo vision,” Image and Vision Computing, pp. 445–456, Mar. 1997.
20. E. Lengyel, “Mathematics for 3D Game Programming and Computer Graphics”, Charles River Media, 2001.

Vision-Based Game Interface Using Human Gesture

Hye Sun Park, Do Joon Jung, and Hang Joon Kim

Department of Computer Engineering, Kyungpook National Univ., Korea
{hspark, djjung, hjkim}@ailab.knu.ac.kr

Abstract. Vision-based interfaces pose a tempting alternative to physical interfaces. Intuitive and multi-purpose, these interfaces could allow people to interact with computer naturally and effortlessly. The existing various vision-based interfaces are hard to apply in reality since it has many environmental constraints. In this paper, we introduce a vision-based game interface which is robust in varying environments. This interface consists of three main modules: body-parts localization, pose classification and gesture recognition. Firstly, body-part localization module determines the locations of body parts such as face and hands automatically. For this, we extract body parts using SCI-color model, human physical character and heuristic information. Subsequently, pose classification module classifies the positions of detected body parts in a frame into a pose according to Euclidean distance between the input positions and predefined poses. Finally, gesture recognition module extracts a sequence of poses corresponding to the gestures from the successive frames, and translates that sequence into the game commands using a HMM. To assess the effectiveness of the proposed interface, it has been tested with a popular computer game, *Quake II*, and the results confirm that the vision-based interface facilitates more natural and friendly communication while controlling the game.

Keywords: Vision-based Game Interface, Gesture Recognition, HMM.

1 Introduction

The quest for intelligent and natural Human Computer Interaction (HCI) has recently received a lot of attention, and influenced the development of various 3D computer games. In particular, if a user can control a 3D computer game using visual, auditory, or simple actions then the user feels more intuitive and interesting during playing a computer game [1-3]. Especially, visual information makes it possible to communicate with computerized equipment at a distance, without any requirement for physical contacts. Also such vision-based interfaces can control games or machines. In this case, a game player feels more intuitive and interesting during playing a computer game.

Various vision based interfaces have been proposed to develop a system for human computer interaction and also a few of them are applied to game or machine. Soshi Iba [4] developed the system that hidden Markov models to spot and recognize gestures captured for controlling robots. Rubine [5] produced a system based on

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 662 – 671, 2006.

statistical pattern recognition techniques for recognizing single-path gestures (drawn with a mouse or stylus) and multiple-path gestures (consisting of a simultaneous path of multiple fingers). Freeman and Weissman [6] presented a television control system using the tracking of hand movements. Krueger [7] developed an experimental interactive environment system called VIDEOPLACE that segments the user's image from a known background, and then builds a composite image of the user's silhouette in a generated environment. Yang et al. [8] developed a gesture based system using a multi-dimensional hidden Markov model (HMM) for telerobotics and human computer interfacing.

In this paper, we present a vision-based interface for detecting and recognizing user gesture in image sequence using a single commercial camera in the *Quake II* game. Previous vision-based interfaces have various constraints such as a specific situation: put on a long-sleeved shirt, a fixed place: stationary lighting and simple background. To solve this problem, it need to more robust vision-based interface. In this paper, the proposed vision-based interface is robust without regard to varying lighting, complex background and various types of cloths (long-sleeved shirt or short-sleeved shirt). Section 2 first of all, introduces such a proposed interface briefly, next explains each modules in detail. In section 3, shows the experimental results of proposed interface. Conclusions are given in section 4.

2 System Overview

In this section, we describe the structure of the proposed interface, the functionality of each process in this structure, and the commands to be recognized. The gesture of a player is recognized as an input of *Quake II* games through our interface. An environment of a proposed interface system is shown in Fig. 1. With a fixed camera, we detect a player's body parts in each frame, we classify player poses from a symbol table, and recognize from pose symbol sequences to player's gestures.

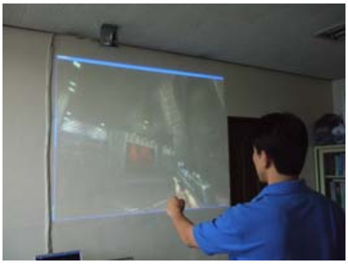


Fig. 1. Environment of the proposed game system

Our interface consists of three main modules: body part localization, pose classification, and gesture recognition. Given the input video sequences, body parts localization involves extraction of important body parts, such as the face and hands, from the each frame, Pose classification involves classifying the extracted input features into predefined pose symbols in a symbol table, and gesture recognition involves recognizing a gesture from among the classified symbols using a single

HMM. *Quake II* receives a recognized gesture command and is operated followed that gesture. The 13 commands are frequently used command in *Quake II*, the commands described in Table 1.

Table 1. Thirteen types of gestures used in *Quake II*

Quake Command	Gesture
WALK FORWARD	Shake a right hand in a forward direction.
BACK PEDAL	Shake a left hand in a backward direction.
ATTACK	Stretch out a right hand in front.
TURN LEFT	Stretch out a left hand to the left.
TURN RIGHT	Stretch out a right hand to the right.
STEP LEFT	Move a right hand down and a left hand to the left.
STEP RIGHT	Move a left hand down and a right hand to the right.
LOOK UP	Move a head to the left.
LOOK DOWN	Move a head to the right.
CENTER VIEW	Stretch out both hands horizontally, then return to the same position.
RUN	Swing arms alternately.
UP/JUMP	Move both hands down and tilt head back.
DOWN/CROUCH	Lift both hands simultaneously.

2.1 Body-Parts Localization

To recognize the user gestures, positions of a face and hands are extracted for each frame of a given image sequence. For this, first of all, we detect a face and hands through three modules: skin-color detection, face detection and hands detection.

2.1.1 Skin-Color Detection

Skin-color information is used to extract the hands and face from the input frame, where the skin-color is represented by an SCT (spherical coordinate transform)- color model, which is reasonably insensitive to variations in illumination and computationally inexpensive, making it simple to implement [9].

The *RGB*-color space is transformed into *SCT*-color space using the following equations:

$$L=\sqrt{R^2+G^2+B^2}\,,\ \angle A=\cos^{-1}\left[\frac{B}{L}\right] \text{ and } \angle B=\cos^{-1}\left[\frac{R}{L\sin(\angle A)}\right], \tag{1}$$

where L is the intensity, and $\angle A$ and $\angle B$ represent the colors independent from the intensity. As such, if a particular color is plotted as a point in *RGB*-space, then the magnitude of the vector for the point is L, the angle between the vector and the blue axis is $\angle A$, and the angle between the red axis and the projection of the vector onto the *RG* plane is $\angle B$. The resulting color space defined by the transformation can be represented by a color triangle [9]. Given skin-color model obtained from experimental results, if the color of a pixel falls within the skin-color distribution range, the pixel is considered as the part of a skin-color area.

However, a background in real world is quiet complex. Thus, detected regions include both regions of user's body, namely skin-color, and noises. For this problem, we use a back ground subtraction method proposed in [10]. Given a background image, we can easily detect a foreground region by subtracting a current image with it. We create the background model BG_i by using the first frame, $F_1(x,y)$. And then we detect the foreground region, FG_i through background subtraction. The foreground region is defined as follows:

$$FG_i(x,y) = \begin{Bmatrix} 1 & |BG_i(x,y) - F_t(x,y)| \geq \theta, \\ 0 & otherwise \end{Bmatrix}, \tag{2}$$

where $BG_i(x,y)$ and $F_t(x,y)$ are the intensity values at the background model and the current frame, respectively. $FG_t(x,y)$ is a binary image that consists of a foreground and a background. The threshold value θ is 30.

We update the background model using current frame F_i and foreground region FG_i. We generate the dilated foreground region FG_{di} which will be used for background updating. We only update the background on the dilated foreground region where $FG_{di} \neq 0$. The background update equation is as follows:

$$BG_{i+1} = (1-w)BG_i + wF_i, \tag{3}$$

where BG_{i+1} is an updated background model, BG_i is current background model, and G_i is a current frame. The weighting factor w is 0.1.

After background subtraction, detected skin-color regions with skin locus have small holes. To eliminate them, we fill up them using a morphological operation. Next we detect positions of face and hands by labeling a region using a connected components analysis. Then the labeled regions are determined face, right hand or left hand as next two sub-sections: face detection and hands detection.

2.1.2 Face Detection

In an image sequence, detecting of the face and hands is difficult because face and hands have a similar property like color information or motion information. For the face detection problem, a process is needed to verify the face candidate region (skin color region) to be a face or non-face. For the face verification, we use the weight vectors of candidate regions in eigenspace. For the dimensionality reduction of the feature space, we project an N-dimensional candidate face image to the lower-dimensional feature space, called eigenspace or facespace [11, 12]. In eigenspace, each feature component accounts for a different amount of the variation among the face images.

To be brief on the eigenspace, let a training set of face images represented by N-dimensional column vector of each image for constructing the facespace be I_1, I_2, I_3,…, I_M. The average of the training set is defined by $A=1/M\Sigma^M_{i=1}I_i$. A new set of vectors with zero mean at each dimension is computed as $\Phi_i = I_i - A$. To produce the M orthogonal vectors that optimally describe the distribution of face images, the covariance matrix is originally computed as

$$C = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T = \mathrm{YY}^T, \tag{4}$$

for $Y = [\Phi_1 \Phi_2 ... \Phi_M]$. Since the matrix *C*, however, is $N \times N$ dimension, determining the *N*-dimensional eigenvectors and *N* eigenvalues is an intractable task. Therefore, for the computational feasibility, instead of finding the eigenvectors for *C*, we calculate *M* eigenvectors $\boldsymbol{v}_k$ and eigenvalues $\boldsymbol{\lambda}_k$ of $[Y^T Y]$, so that $\boldsymbol{u}_k$, a basis set is computed as

$$u_k = \frac{Y \times v_k}{\sqrt{\lambda_k}}, \tag{5}$$

for *k*=1, …, *M*. Of the *M* eigenvectors, the *M'* significant eigenvectors are chosen as those with the largest corresponding eigenvalues. For *M* training face images, the feature vectors $W_i = [w_1, w_2, \ldots, w_{M'}]$ are calculated as

$$w_k = u_k^T \Phi_i, \quad k = 1, \ldots, M. \tag{6}$$

To verify the candidate face region to be a face or non-face, the candidate face regions are also projected into the trained eigenspace using Eq. (6). The projected regions are verified using the minimum distances of the detected regions with the face cluster and the non-face cluster according to Eq. (7).

$$\min(\| W_k^{candidate} - W_{face} \|, \| W_k^{candidate} - W_{nonface} \|), \tag{7}$$

where $W_k^{candidate}$ is the *k*th candidate face region in trained eigenspace, and W_{face} and $W_{nonface}$ are the center coordinate of the face cluster and non-face cluster in trained eigenspace respectively. Then, the Euclidean distance measure is used. If there are more than two verified faces in an image frame, we select a biggest one as a face.

2.1.3 Hands Detection

We detect right and left hands using heuristic rules. First, hands are the two biggest regions of skin-color regions except a face region. Next, left one is left hands and right one is right hand under a restriction: the hands are not crossing in no case.

However, when the user put on the short-sleeved shirt, it is hard to detect hands. Thus, we add to the following heuristic rules: the shape of hand (a fist) is circle and is bigger than the detected arm. But the forearm is bigger than a fist when user bends his arm, so we considered human physical character and gesture character. Thus it followed as: if the extracted hand included arm region is closely a face then it searches hand in region of upper half in the detected region. Otherwise it searches hand in region of lower half in the region.

Given search-space in detected hand region, we finally determine that hand is the region has maximum score that satisfies the Eq. 8.

$$score = magnificat\,ion\ of\ circle \times \frac{a\ number\ of\ skin\ color\ pixels\ inside\ the\ hand\ circle}{a\ number\ of\ pixels\ \ inside\ the\ hand\ circle} \tag{8}$$

The hand circle is extended by 5 different magnifications to find potential locations for the hand.

2.2 Pose Classification

The body parts positions of a frame are classified to a pose symbol that have a smallest norm to the given body parts positions in a symbol table predefined. The norm is defined as a Euclidean distance in 6D-vector space of the body positions. The symbol table is trained off-line using *K*-means clustering method. For this, we extract poses which are represented the gesture per each gesture. In experiment results, *K* is 64 and the number of poses is selected as 23 under consideration that some poses which are similar to the others are same one.

2.3 Gesture Recognition

To recognize a gesture, we take a continuous stream of pose symbols. Every-time a symbol is given, a gesture recognizer determines whether the user is performing one of the thirteen gestures predefined above, or not. Usually HMMs are working on isolated or pre-segmented sequences of input symbol sequences. For this, it needs to find an automatic way of segmenting the sequences. But in reality, it is hard to find an easy way to find this segmentation. So we propose a new architecture of a HMM that takes a continuous stream of pose symbols with automatically segmenting and recognizing abilities. The proposed HMM is a single HMM modeled the thirteen gestures. Given a continuous stream of images as an input, the HMM starts with initial state probabilities $s^0 = \{s_k^0\}$ and continuously updates its state probabilities with each symbol of that stream as shown in Eq. 9.

$$\bar{s}_n^t = \sum_{k=0}^{k-1} (s_k^{t-1} \times a_{kn}) \times b_{np} \ , \ s_n^t = \frac{\bar{s}_n^t}{\sum_{k=0}^{k-1} \bar{s}_k^t} \qquad (9)$$

• *S* = {*sk*}: A vector of state probabilities, *sk* denotes a state probability of state *k*.	
• *K* = the number of states,	• *M* = the number of poses.
• *A*={*aij*}: An *K* × *K* matrix for the state transition probability distributions where *aij* is the probability of making a transition from state *si* to *sj*.	
• *B*={*bij*}: An *K* × *M* matrix for the observation symbol probability distributions where *bij* is the probability of emitting pose symbol *vk* in state *si*.	

Then, the HMM has a few states that distinguish the path related with each gesture. So when the state has higher state probability than predefined threshold, a gesture is detected and recognized.

3 Experimental Results

The experimental results showed that vision-based interface allowed that gamer feel more natural and intuitive during he or she play a game. The proposed interface was implemented on a *PentiumIV-2.8GHZ PC* using *Visual C++* language. In the proposed interface, we improve various restrictions in [4,6,13]. In sub-section, we describe each module's results in detail.

3.1 Skin-Color Detection Result

To demonstrate the robustness of the proposed skin-color detection, tests were performed using sequences collected on different days, at different times, under varying lighting environments from four different individuals. To obtain the skin-color model, skin-color regions were extracted from 100 test data, the color distribution of these regions investigated, and the skin-color finally represented by the following parameters: $\angle A$ is 1.001~1.18, $\angle B$ is 0.82 ~ 0.97, and L is 170 ~ 330.

3.2 Face Detection Result

To detect the user's face in skin-color regions, we experiment with twenty different test sequences, and the train to sets consisted of six individuals with five different face orientations. Fig. 2 shows some of the training images used to construct the eigenspace.

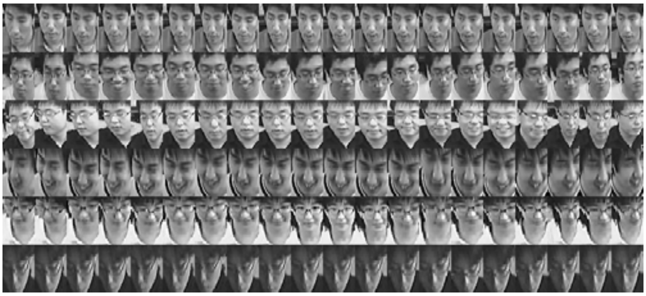


Fig. 2. The parts of the training images for eigenspace construction

An analysis of the image sets captured during the experiment revealed that the face detection was 91.0% correct on average.

$$\textit{Face detection rate} = \frac{\textit{Number of correctly detected images as faces}}{\textit{Number of images detected as true faces}} \tag{10}$$

$$\textit{Non\_Face detection rate} = \frac{\textit{Number of correctly detected images as non\_faces}}{\textit{Number of images detected as non\_faces}} \tag{11}$$

In Table 2, the face detection success rates represent when the face regions were detected as faces and the non-face regions detected as non-faces.

Table 2. The face detection success rates

Users	Face	Non_Face	Total
1	93.8%	91.8%	92.8%
2	90.9%	88.3%	89.6%
3	91.3%	90.8%	91.0%
4	90.0%	91.0%	90.5%
5	92.5%	90.0%	91.3%
6	91.0%	89.9%	90.5%
Total	91.6%	90.4%	91.0%

3.3 Hands Detection Result

Detected hand region is represented by circle. To find potential locations for the hand, the circle is extended by 5 different magnifications. The circle is extended by 1×radius, 2×radius, 3×radius, 4×radius, 5×radius, the radius is 3 pixels in our experiment. Fig. 3 (a) and (b) show hands detection result with the different magnifications of the circle for potential hands locations. The error is occurred when it is failed to detect face or the hand region is not dominant. Fig. 3 (c) shows an example of detection error. An analysis of the image sets captured in the experiments results that the hands detection was 95.1% correct on average.

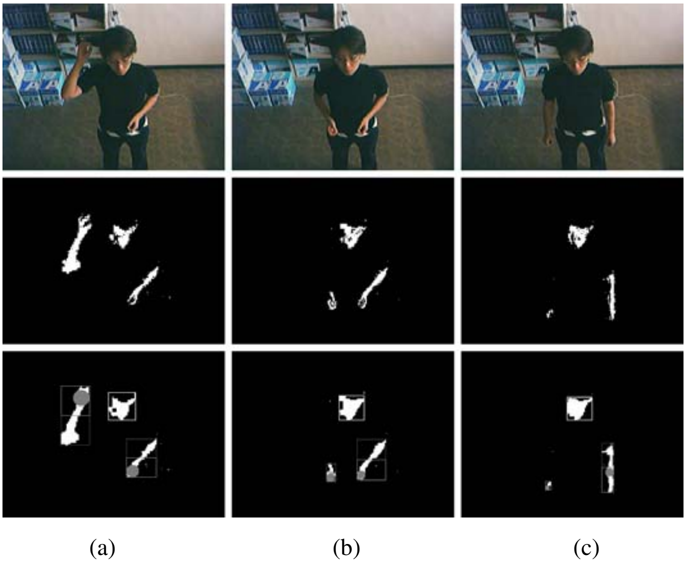


Fig. 3. Hands detection results. Each example contains an original image (first row), skin color images (second row), and overlaid with detected hands (third row: in the third row, a bounding box with a white solid line is detected face region, bounding boxes with a gray dotted line are skin-color region except face region, bounding boxes with a gray solid line are hand searching region according to the human physical character).

3.4 Gesture Recognition Results

The performance of the proposed HMM recognizer was analyzed using a total of 1300 frames of thirteen gestures performed many times by different individuals. The results are summarized in Table 3 based on the detection and reliability ratio used in [14], defined using the following Eqs. (11) and (12). The experimental results showed a 93.5% accuracy.

$$Detection = \frac{Correctly\ recognized\ gestures}{The\ number\ of\ input\ gestures} \tag{12}$$

$$Reliability = \frac{Correctly\ recognized\ gestures}{The\ number\ of\ input\ gestures\ +\ the\ number\ of\ insertion\ errors}, \tag{13}$$

where '*Insert*' means the errors when the recognizer detected non-existing gestures. Plus, '*Delete*' and '*Substitute*' represent when the recognizer failed to detect a gesture and misclassified a gesture, respectively.

Table 3. Gesture-recognition results

Command	Num. of Gestures	Results					
		Insert	Delete	Substitute	Correct	Detection(%)	Reliability(%)
WALK FORWARD	327	0	27	0	300	91.74	91.74
BACK PEDAL	89	0	8	0	81	91.01	91.01
TURN LEFT	117	0	0	0	117	100	100
TURN RIGHT	132	0	0	0	132	100	100
STEP LEFT	83	0	0	0	83	100	100
STEP RIGHT	92	9	0	0	92	100	91.09
ATTACK	236	0	17	0	219	92.80	92.80
CENTER VIEW	52	0	3	0	49	94.23	94.23
UP/JUMP	62	0	2	0	60	96.77	96.77
DOWN/ CROUCH	43	0	4	0	39	90.70	90.70
RUN	32	0	2	8	22	68.75	68.75
LOOK DOWN	24	0	3	0	21	87.50	87.50
LOOKUP	27	0	3	0	24	88.89	88.89
Total	1316	9	69	8	1239	94.15%	93.50%

4 Conclusions

In this paper, we proposed a vision-based interface and applied to a gesture based game system for the *Quake II*. In implemented game system, the proposed method showed 93.5% reliability, which was enough to enjoy the game *Quake II* using gestures, instead of the existing interface such as keyboard and mouse. Also the proposed method is robust in varying environments such as varying lighting, complex background and user putted on various clothes. Therefore the user can enjoy playing game within natural real-environment.

References

1. Sara Kiesler and Pamela Hinds: Human-Robot Interaction, Journal of HCI: Special Issue on HCI, Vol.19 No. 1 & 2 (2004)
2. Maja Pantic and Leon J. M. Rothkrantz: Toward an Affect-Sensitive Multimodal Huamn-Computer Interaction, Proceedings of the IEEE Vol. 91 No. 9 (2003)
3. B. A. MacDonald and T. H. J. Collett: Novelty Processing in Human Robot Interaction, Symposium of one-day trans-disciplinary inter-media in Elam School of Fine Arts at the University of Auckland (2004)
4. S. Iba, J. M. V. Weghe, C. J. J. Paredis and P. K. Khosla: An Architecture for Gesture based Control of Mobile Robots, Intelligent Robots and Systems Vol. 2 (1999)
5. D.H. Rubine: The automatic recognition of gesture, Ph.D dissertation, Computer Science Department, Carnegie Mellon University December (1991)
6. W.T. Freeman and C.D. Weissman: Television control by hand gestures, Int. Workshop of Race and Gesture Recognition (1995) 179-183
7. M.W. Krueger: Artificaial Reality II, Addison-Wesley Reading MA (1991)
8. J. Yang, Y. Xu and C.S. Chen: Gesture Interface: Modeling and Learning, Robotics and Automation (1994) 1747-1752
9. J.Hymes, M.W. Powell, R. Murphy: Cooperative navigation of micro-rovers using color segmentation, IEEE International Symposium on Computational Intelligence in Robotics and Automation, (1999) 195-201
10. F. S. Chen, C. M. Fu and C. L. Hyang: Hand Gesture Recognition using a Real-time Tracking Method and Hidden Markov Models, Image and Vision Computing Vol. 2 (2003)
11. Turk, M., Pentland, A.: Face Recognition Using Eigenfaces. IEEE Proceedings CVPR '91, (1991) 586 -591
12. Guan, A.X., Szu, H.H.: A local face statistics recognition methodology beyond ICA and/or PCA. International Joint Conference on Neural Network, (1999) 1016 -1021
13. H. Kang, C.W. Lee and K. Jung: Recognition-based gesture spotting in video games, Pattern Recognition Letters Vol. 25 (2004) 1701-1714
14. H. K. Lee and J. H. Kim: An HMM-based Threshold Model Approach for Gesture Recognition, IEEE Transaction on Pattern Analysis and Machine Intelligence Vol. 21 (1999)

Terminal Phase Vision-Based Target Recognition and 3D Pose Estimation for a Tail-Sitter, Vertical Takeoff and Landing Unmanned Air Vehicle

Allen C. Tsai, Peter W. Gibbens, and R. Hugh Stone

School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney,
N.S.W., 2006, Australia
{allen.tsai, pwg, hstone}@aeromech.usyd.edu.au

Abstract. This paper presents an approach to accurately identify landing targets and obtain 3D pose estimates for vertical takeoff and landing unmanned air vehicles via computer vision methods. The objective of this paper is to detect and recognize a pre-known landing target and from that landing target obtain the 3D attitude information of the flight vehicle with respect to the landing target using a single image. The Hu's invariant moments' theorem is used for target identification and parallel lines of the target shape are investigated to obtain the flight vehicle orientation. Testing of the proposed methods is carried out on flight images obtained from a camera onboard a tail-sitter, vertical takeoff and landing unmanned air vehicle.

Keywords: Tail-sitter vertical takeoff and landing unmanned air vehicle, computer vision, moment invariants, vision-based pose/attitude estimation, target identification/detection, parallel lines, vanishing points, perspective transformation and vision-based autonomous landing.

1 Introduction

Using cameras as sensors to perform autonomous navigation, guidance and control for either land or flight vehicles via computer vision has been the latest developments in the area of field robotics.

This paper focuses on how a tail-sitter Vertical Takeoff and Landing (V.T.O.L.) Unmanned Air Vehicles (U.A.V.) can use visual cues to aid in navigation or guidance, focusing on the 3D attitude estimation of the vehicle especially during the terminal phase: landing; where an accurate tilt angle estimate of the vehicle and position offset from the landing target is required to achieve safe vertical landing. Work in this particular field focusing on state estimation to aid in landing for V.T.O.L. U.A.V. has been done by a number of research groups around the world. Work of particular interest is the U.S.C. A.V.A.T.A.R. project[1]; landing in unstructured 3D environment using vision information has been reported but only the heading angle of the helicopter could be obtained from those vision information. Work done by University of California, Berkeley[2] focused on ego-motion estimation where at least two or more images are

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 672 – 681, 2006.

required to determine the 3D pose of a V.T.O.L. U.A.V. Yang and Tsai[3] have looked at position and attitude determination of a helicopter undergoing 3D motion using a single image, but did not present a strategy for target identification. Amidi and Miller [4] used a visual odometer but the visual data could not be used as stand alone information to provide state estimation in 3D space to achieve landing.

This paper will depict how from a single image, target identification is achieved as a V.T.O.L. U.A.V. undergoes 3D motion during hover, just before landing, as well as being able to ascertain 3D pose from the appearance of the landing target which is subject to perspective transformation. The 3D attitudes of the vehicle are determined by investigating parallel lines of the landing target. Target identification is achieved through the calculations of Hu's invariant moments of the target.

Paper Outline: Section 2 will discuss in detail the image processing techniques undertaken, set up of the video camera on the "T-Wing" and the design of the landing pad; section 3 looks into the mathematical Hu's invariant moments' theorem applied in order to detect and recognize the landing target, section 4 presents the mathematical techniques used to carry out 3D pose estimation, and section 5 presents and discusses the results for the strategies proposed tested on flight images taken from a U.A.V.; and lastly for Section 6, conclusion and directions for future work are drawn.

2 Vision Algorithm

The idea of the vision algorithm is to accurately maintain focus on an object of interest, which is the marking on the landing pad (here on after will be referred to as the landing target), by eliminating objects that are not of interest. The elimination of unwanted objects is achieved by a series of transformation from color to a binary image, filtering and image segmentations. In this section the image acquisition hardware set-up and the set up of the landing pad is firstly introduced and following that is the details of the vision algorithm.

2.1 Image Acquisition Hardware Set-Up

The capital block letter "T" is used as the marking on the landing pad to make up the landing target. The idea behind using a "T" is that it is of only one axis of symmetry whereas the more conventional helicopter landing pads, either a circle or the capital block letter of "H", have more than one axis of symmetry. The one axis symmetry will allow uniqueness and robustness when estimating the full 3D attitudes and/or positions of the flight vehicle relative to Ground Coordinate System which is attached to the landing target.

The camera used was a C.C.T.V. camera, the imaging sensor was of a 1/3″ Panasonic Color CCD type. The resolution is of 737 horizontal by 575 vertical pixels with a field of view of 95° by 59° and records at 25 Hz. Recording of the images of the on-board camera was achieved by transmission using a 2.4GHz wireless four channel transmitter to a laptop computer where a receiver is connected. The camera was calibrated using online calibration toolbox [5]. The test bed used for the flights was the T-Wing [6] tail-sitter V.T.O.L. U.A.V. which is currently under research and development at the University of Sydney. The set up of the camera on the flight vehicle is shown on the following simulation drawing and photos:

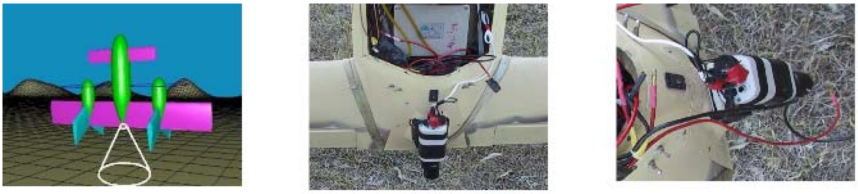


Fig. 1. Simulation drawing and photos of the camera set-up on the T-Wing

2.2 Image Processing Algorithm

The low level image processing for the task of target identification and pose estimation requires a transformation from the color image to a gray-scale version to be carried out first. This basically is the elimination of the hue and saturation information of the RGB image while maintaining the luminance information.

Once the transformation from color to gray-scale is achieved, image restoration, i.e. the elimination of noise, is achieved through a nonlinear spatial filter, median filter, of mask size [5 × 5] is applied twice[7] over the image. Median filters are known to have low pass characteristics to remove the white noise while still maintaining the edge sharpness, which is critical in extracting good edges as accuracy of attitude estimation is dependent on the outcome of the line detection.

After the white noise is removed, a transformation from gray-scale to binary image is required by thresholding at 30% from the maximum intensity value, in an effort to identify the object of interest: the landing target, which is marked out in white. Despite that, the fact of sun's presence and other high reflectance objects that lie around the field where the experiment was carried out; it is not always possible to eliminate other white regions through this transformation.

Image segmentation and connected component labeling was carried out on the images in an attempt to get rid of the objects left over from gray-scale to binary transformation that are of no interest. These objects are left out by determining that their area is either too big (reflectance from the ground due to sunlight) or too small (objects such as other markings). The critical area values for omitting objects are calculated based on the likely altitude and attitudes of the flight vehicle once in hover, which is normally in between two to five meters and ±10º. Given that the marking "T" is of known area and geometry the objects to be kept in interest should have an area within the range 1500 to 15,000 pixels after accounting for perspective transformation. This normally leaves two or more objects still present in the image, which was the case for about 70% of the frames captured. The Hu's Invariant Moments theorem is then applied as a target identification process in an attempt to pick out the correct landing target.

Once the landing target is determined the next stage is to determine the edges of the block letter "T" in order to identify the parallel lines which are to be used later on for attitude estimation. The edge detection is carried out via the Canny edge detector [8];

the line detection is carried out using the Hough transform [8]. The following figure shows the stages of image processing:

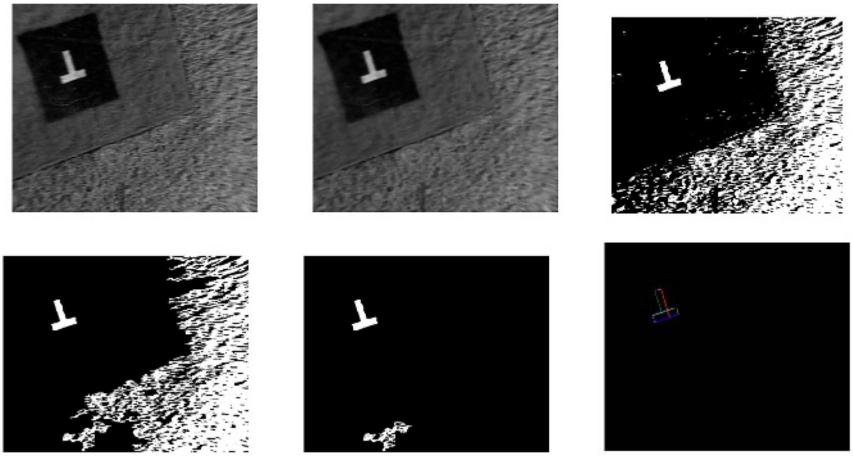


Fig. 2. Image processing stages (from left to right and down): 1st the grayscale transformation, median filtering, binary transformation, rejection of small objects, rejections of large objects, and finally target identification with line detection

3 Target Identification

The landing target identification procedure is accomplished by using the geometric properties of the target. This involves investigating the moment of inertia of the target shape. Hu's invariant moments[9] are known to be invariant under translation, rotation and scaling in the 2D plane. This feature is very well suited to tasks associated with identifying landing targets. The following equation represents the moments of a 2D discrete function:

$$m_{pq} = \sum_{i} \sum_{j} i^{p} j^{q} I(i, j) \tag{1}$$

where $(p + q)$ represent the order of the moments and $I(i,j)$ is the intensity of an image. The indexes i, j corresponds to the image plane coordinate axes x and y indexes respectively.

The central moments are the moments of inertia defined about the centre of gravity and are given by the following equation:

$$\mu_{pq} = \sum_{i} \sum_{j} (i - \overline{x})^{p} (j - \overline{y})^{q} I(i, j) \tag{2}$$

where the indexes have the same meaning as in equation (1), $\overline{x}$ and $\overline{y}$ are the centroids of the target shape.

The normalized central moment is defined as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{3}$$

Where $\gamma = (p + q)/2$, for $p + q = 2, 3, ...$

The first four orders of invariant moments can be determined by the normalized central moments, they are as follows:

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11} \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \end{aligned} \tag{4}$$

In this paper all these four orders of invariant moments were tracked to carry out identification of the landing target allowing for perspective distortion. An object is considered to be the target if the sum of errors for all four orders is the minimum of all the objects.

As the tail-sitter V.T.O.L. U.A.V. approach the landing phase, the flight vehicle will always undergo 3D motion, the invariant moment method described above is only known to be invariant under 2D scaling, translation and rotation. Therefore it is critical to investigate all facets of the invariant moments. As proven by Sivaramakrishna and Shashidharf [10], it is possible to identify objects of interest from even fairly similar shapes even if they undergo perspective transformation by tracking the higher order moments as well as the lower order ones. This method is applied later on to determine the correct target, i.e. target identification.

4 State Estimation

Due to the inherent instability with V.T.O.L. U.A.V. near the ground during landing, it is necessary to be able to accurately determine the 3D positions and 3D attitudes of the flight vehicle relative to the landing target in order to carry out high performance landing. The information given by the parallel lines of the target is one of vision-based navigation techniques used to obtain high integrity estimation of the 3D attitudes of the flight vehicle. Because of the landing pad marking: ”T”, there are two sets of nominally orthogonal parallel lines that are existent, the two sets of orthogonal parallel lines each containing only two parallel lines are named A and B. The parallel lines in set A and set B correspond to the horizontal and vertical arm of the “T” respectively.

4.1 Coordinate Axes Transformation

Before establishing the relationship of parallel lines to determine the vehicle attitude, the transformations between several coordinate systems need to be ascertained. The first coordinate system to be defined is the image plane denoted as the I.C.S. (Image Coordinate System), and then the camera coordinate system (C.C.S.) which represents the camera mounted on the flight vehicle. Lastly the flight vehicle body axes denoted as the vehicle coordinate system (V.C.S.). The line directions of the three flight vehicle body axes in the C.C.S. are pre-known and denoted as d_{normal}, $d_{longitudinal}$ and $d_{lateral}$. To describe the relative orientation between the landing target and the flight platform, the global coordinate system (G.C.S.) is also required. The axes of the G.C.S. are denoted as X, Y and Z. The X axis is parallel with the horizontal bar of the block letter "T" and pointing to the "T"'s right; the Y axis is parallel with the vertical bar of the "T" and a positive down direction. These two axes are situated at the centre point where the vertical and horizontal bars meet rather than the conventional centre of gravity of the shape. The Z axis is therefore pointing up according to the right hand rule. The transformation from the C.C.S. to the V.C.S. amounts to an axis re-definition according to:

$$\tilde{X}_{V.C.S.} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \times \tilde{X}_{C.C.S.} \tag{5}$$

4.2 Flight Vehicle Orientation Estimation Via Parallel Lines Information

Moving onto the application of parallel lines of the target shape to deduce 3D attitudes of the flight vehicle; it is well known that a set of 3D parallel lines intersect at a vanishing point on an image plane due to perspective transformation. The vanishing point is a property that indicates the 3D line direction of the set of parallel lines [11].

Considering a 3D line "L" represented by a set of points:

$$L = \{(x, y, z) \mid (x, y, z) = (p_1, p_2, p_3) + \lambda(d_1, d_2, d_3)\, for\ real\ \lambda\} \tag{6}$$

The line L passes through the point (p_1, p_2, p_3) and has the line directions (d_1, d_2, d_3). The image plane point of a 3D point on the line L can be written as:

$$\begin{aligned}(u, v) &= (f\times x/z, f\times y/z) \\ &= [f\times(p_1 + \lambda d_1)/(p_3 + \lambda d_3),\ f\times(p_2 + \lambda d_2)/(p_3 + \lambda d_3)]\end{aligned} \tag{7}$$

where f is the focal length of the camera and λ is the line length or parameter.

A vanishing point (u_∞, v_∞) can be detected if $\lambda \rightarrow \infty$ and $d_3 \neq 0$; that is:

$$(u_\infty, v_\infty) = [f\times d_1/d_3, f\times d_2/d_3] \tag{8}$$

The direction of the line "L" can now be uniquely determined from the above equation to be:

$$(d_1, d_2, d_3) = (u_\infty, v_\infty, f)/\sqrt{u_\infty^2 + v_\infty^2 + f^2} \tag{9}$$

With the above theory applied to the problem of pose estimation; the line directions of the X axis, which corresponds to the horizontal bar of the "T", and the Y axis, corresponding to the vertical bar of the "T", in the C.C.S. can firstly be determined. The line directions are denoted as d_x, d_y and d_z, where the line direction of d_z is determined by the cross product of d_x and d_y.

Once the directions of the G.C.S. axes in the C.C.S. are determined the flight platform's attitudes with respect to the landing target can be determined via the following, which takes into account the transformation from C.C.S. to V.C.S.:

$$\begin{aligned} \cos\alpha &= (d_x \cdot d_{Long.})/(\|d_x\| \times \| d_{Long.}\|), \\ \cos\beta &= (d_y \cdot d_{Lat.})/(\|d_y\| \times \| d_{Lat.}\|), \\ \cos\gamma &= (d_z \cdot d_{Normal.})/(\|d_z\| \times \| d_{Normal.}\|). \end{aligned} \tag{10}$$

d_x, d_y and d_z are the direction cosines of the G.C.S. axes in the C.C.S. α, β, and γ are the angles of the G.C.S. axes in the V.C.S., which correspond to the roll, pitch and yaw angles of the flight vehicle respectively. The following figure shows the relations between the C.C.S., G.C.S. and I.C.S., and the line direction of the vanishing point in C.C.S.

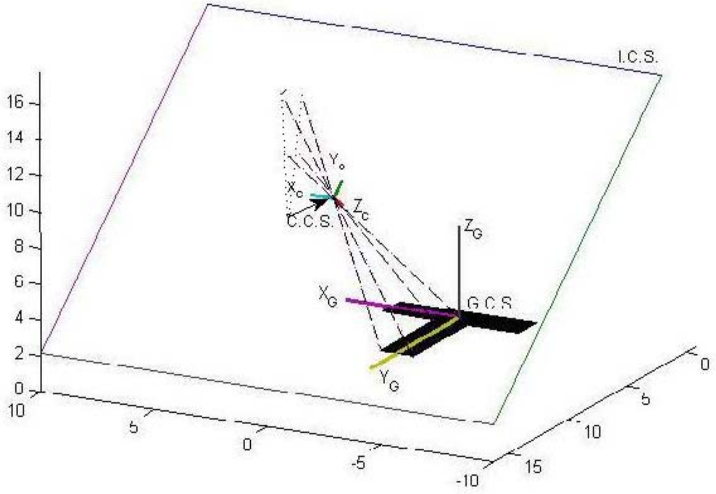


Fig. 3. Schematic diagram of the relations between I.C.S., C.C.S., and G.C.S., and the line direction of G.C.S. Y axis in the C.C.S.

5 Experimental Results

Flight images taken during the flight testing of the "T-Wing", a tail-sitter V.T.O.L. U.A.V., were used to test the accuracy, repeatability and computational efficiency of the fore mentioned theories of target recognition and 3D attitude estimation. Flight images are extracted from a 160 seconds period of the flight vehicle hovering over the target with the target always kept insight of the camera viewing angles. Figure 4 shows the three angles between the V.C.S. and the G.C.S. as the flight platform pitches, yaws and rolls during the hover.

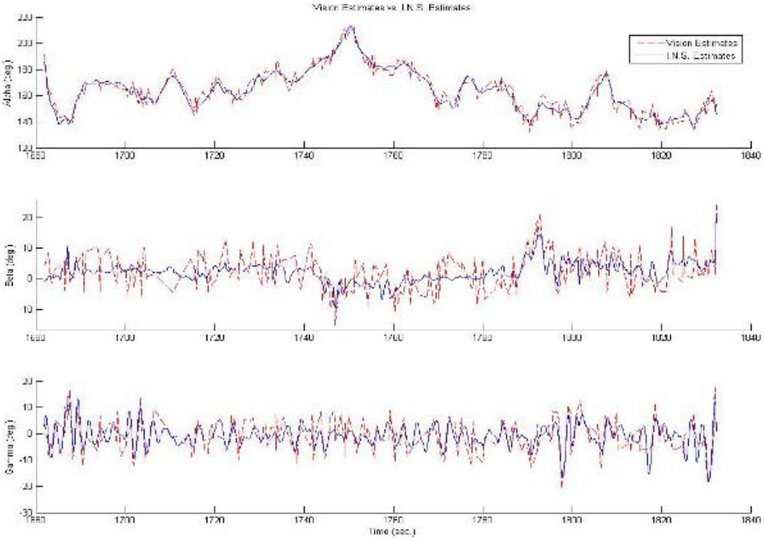


Fig. 4. Plots of the Alpha, Beta and Gamma angles ascertained from vanishing points and the respective roll, pitch and yaw angles of the vehicle

The estimated attitudes were compared with filtered estimates from a NovAtel RTK G.P.S. unit, accuracy of down to 2cm and a Honeywell Ring-Laser Gyro. of 0.5° accuracy. The errors from the attitude estimates from the parallel line information compared favorably with R.M.S. errors of 4.8° in alpha, 4.2° in beta and 4.6° in gamma. This range of error is deemed good by other work's standard[12].

Figure 5 shows the computed 1st, 2nd, 3rd and 4th order invariant moments of the landing target, "T", from the images captured during the vehicle hover. Knowing that the invariant moments are only "invariant" under 2D translation, scaling and rotation, all four orders of the invariant moments were tracked. The results show that the error of the invariants moments computed as the flight vehicle hovers above the target are larger than the errors associated with 2D motion; but by tracking of the first four orders of invariant moments, the target always had the smallest total normalized error (sum of all four percentage discrepancy to true value obtained from noiseless images divided by four) than other objects remaining in the images. The first four orders of invariant moments shows that the normalized errors were greater for the period where the vehicle undergoes greater pitching and yawing moments than periods where the vehicle is almost in a perfect vertical hover mode. There were images where noise was a big issue but nevertheless by tracking all the four orders of invariant moments, the landing target could still be distinguished.

With regard to computational time, the filtering and thresholding of the images took approximately 11.92% of the time; component labeling and segmentation took around about 54.99%; attitude estimation algorithm needed 4.933% and the invariant moments' calculations required 28.16% of the computational time when dealing with two objects.

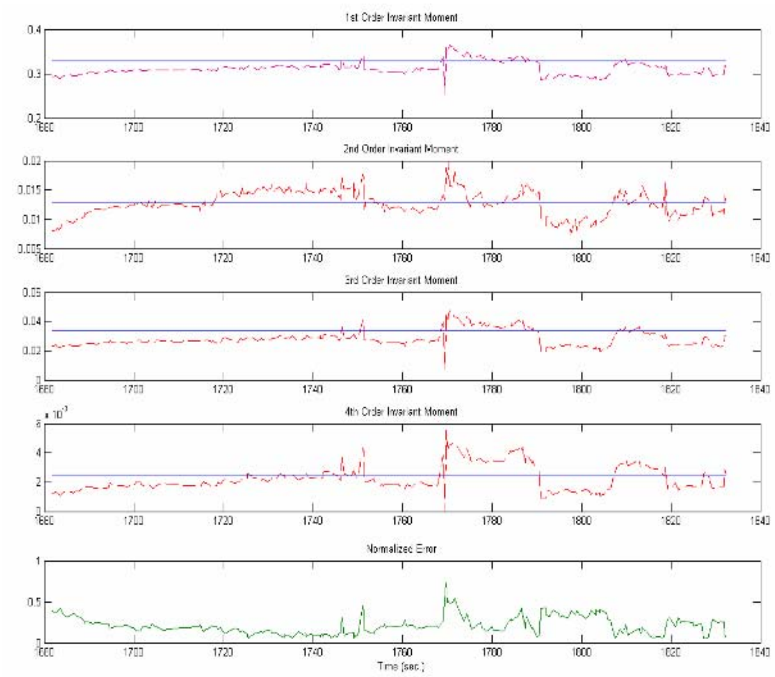


Fig. 5. The computed 1st, 2nd, 3rd and 4th order invariant moments (in red --) compared with true values (in blue -); and the total normalized error

6 Conclusion

In this paper, an algorithm to accurately identify landing targets and from those landing targets, using computer vision techniques, to obtain attitudes estimates of a tail-sitter V.T.O.L. U.A.V. undergoing 3D motion during the hover phase is presented. This method of 3D attitude estimation requires only a single image which is a more computationally effective technique than motion analysis which requires processing of two or more images. This paper also presented techniques of accurately determining landing targets via the invariant moments' theorem while an air vehicle is undergoing 3D motion. The results show a good accuracy with target detection and pose estimation with previous other work in this field. Further developments of these techniques in the future can possibly see autonomous landing of manned helicopter onto a helipad and commercial aircrafts autonomously detecting runways during landing.

A major issue requiring investigation is the estimation of attitude when the landing target is only partially visible in the image plane. We intend in the future to integrate the attitude, position and velocities estimates, to act as guidance information, with the control of the "T-Wing" especially during landing.

Acknowledgements. J. Roberts for his technical expertise with setting up of the camera system and the wireless transmission for recordings.

References

[1] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle.", IEEE International Conference on Robotics and Automation, ICRA'02 Proceedings, Washington, DC, 2002.

[2] O. Shakernia, R. Vidal, C. S. Sharp, Y. Ma, and S. Sastry, "Multiple view motion estimation and control for landing an unmanned aerial vehicle.", IEEE International Conference on Robotics and Automation, ICRA'02 Proceedings, Washington, DC, 2002.

[3] Z. F. Yang and W. H. Tsai, "Using parallel line information for vision-based landmark location estimation and an application to automatic helicopter landing," *Robotics and Computer-Integrated Manufacturing*, vol. 14, pp. 297-306, 1998.

[4] T. K. O. Amidi, and J.R. Miller, "Vision-Based Autonomous Helicopter Research at Carnegie Mellon Robotics Institute 1991-1997.", American Helicopter Society International Conference, Heli, Japan, 1998.

[5] Z. Zhang, "Flexible Camera Calibration by viewing a plane from unknown orientation.", International Conference on Computer Vision (ICCV'99), Corfu, Greece, pp. 666-673, September 1999

[6] R. H. Stone, "*Configuration Design of a Canard Configuration Tail Sitter Unmanned Air Vehicle Using Multidisciplinary Optimization.*" PhD Thesis, University of Sydney, Australia, 1999.

[7] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*: Pearson Prentice Hall, Upper Saddle River, N.J., 2004.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2 ed., Pearson Prentice Hall, Upper Saddle River, N.J., 2002.

[9] M. Hu, "Visual Pattern Recognition by Moment Invariants.", *IRE Transactions on Information Theory*, 1962.

[10] R. Sivaramakrishna and N. S. Shashidharf, "Hu's moment invariants: how invariant are they under skew and perspective transformations?", IEEE WESCANEX 97: Communications, Power and Computing. Conference Proceedings, 1997.

[11] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. II: Addison-Wesley, 1993.

[12] C. S. Sharp, O. Shakernia, and S. S. Sastry, "A vision system for landing an unmanned aerial vehicle.", IEEE International Conference on Robotics and Automation, ICRA '01 Proceedings., 2001.

Detection of Traffic Lights for Vision-Based Car Navigation System

Hwang Tae-Hyun, Joo In-Hak, and Cho Seong-Ik

Telematics.USN Research Division, ETRI
161 Kajeong-dong, Yuseong-gu, Daejeon, Republic of Korea
{hth63339, ihjoo, chosi}@etri.re.kr

Abstract. A recent trend of car navigation system is using actual video captured by camera equipped on a vehicle. The video-based navigation systems displays guidance information overlaid onto video before reaching a crossroad, so it is essential to detect where the crossroads are in the video frame. In this paper, we suggest a detection method for traffic lights that is used for estimating location of crossroads in image. Suggested method can detect traffic lights in a long distance, and estimates pixel location of crossroad that is important information to visually represent guidance information on video. We suggest a new method for traffic light detection that processes color thresholding, finds center of traffic light by Gaussian mask, and verifies the candidate of traffic light using suggested existence-weight map. Experiments show that the detection method for traffic signs works effectively and robustly for outdoor video and can used for video-based navigation system.

1 Introduction

As the advent of telematics as a newest field of information technology, car navigation systems (CNS) have been developed as a key application in telematics field, and have got much interests of the users and demands of service providers. CNSs have been introduced in the form of PDA-based system that is based on 2-dimensional digital map and route planning/guidance functions. Recent service providers have begun to support 2.5D and 3D display.

However, navigation systems till now have limited representation power for real world, especially in case of confusing crossroad or junction. Such defect may make drivers get lost or even cause traffic accidents. In this respect, navigation systems are now aiming at more realistic representation of real world. A newest direction of navigation system is to use actual video captured by camera equipped on a vehicle. The examples of early video-based systems are INSTAR of Siemens[1] and VICNAS of Kumamoto University[2]. Such systems intend to enhance drivers' understanding about real world by capturing real-time video and displaying navigation information (such as directed arrows, symbols, and icons) *onto* it. Because video is more perceptible than conventional map (either paper or digital), such video-based navigation systems are expected to reduce users' visual distractions while driving and to make navigation guidance easier

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 682–691, 2006.

and more convenient. To make the video-based navigation possible, navigation information should be overlaid accurately onto video. An approach is AR (Augmented Reality)-styled display, but such approach requires very accurate position/attitude of vehicle as well as geographic topography, for which high-cost sensors (including camera) and large-volumed database are needed.

To solve the problem, we suggest a computer vision technology that processes input video and helps to overlay navigation information onto the video. Considering that the most important function of a navigation system is to guide direction before a crossroad is reached, we focus on *detecting the location of crossroad in image*, as a key to our vision-based approach. However, practically speaking, the guidance should be done in a long distance before a crossroad. Directly detecting crossroad in a long distance is quite difficult (even human's eye may not identify it), so we detect traffic lights rather than crossroad itself. Assuming that there are traffic lights at most of crossroads, we can estimate pixel location of a crossroad if a traffic light is detected. Many detection methods have been suggested for road facilities including traffic lights [3][4][5], but most of the methods are not applicable to the vision-based navigation systems because they cannot detect traffic lights in a long distance. To effectively and robustly detect traffic lights for our goal, we suggest a new detection method that can detect traffic lights in a long distance.

The remainder of this paper is organized as follows. Section 2 provides a detection method for traffic lights. Section 3 describes how the detection functions are worked with car navigation system. In Section 4 we present the experimental results, and Section 5 concludes our suggestions.

2 Detection of Traffic Lights

The method suggested in this paper detects traffic lights by an algorithm shown in Fig.1. Each process is executed sequentially, as described below.

2.1 Traffic Lights Detection

Color thresholding. In this paper we use HSI color model to analyze color of traffic lights. It is a color model based on human vision, and many researches have used color thresholding with the HSI color model [3]. It can be used as a measure for distinguishing predefined colors of traffic lights. We refer an official guideline for colors of traffic lights, published by National Police Agency of Korea, the competent authority of management of traffic lights[6]. In this guideline, each color of traffic lights is defined as an area in CIE chromaticity diagram, in which yellow and red are in almost same area. Therefore, we use same color information when detecting yellow and red traffic lights.

After transforming RGB color model into HSI color model [7], we execute thresholding with regard to red and green color components of CIE chromaticity to get candidate colors of traffic lights, as

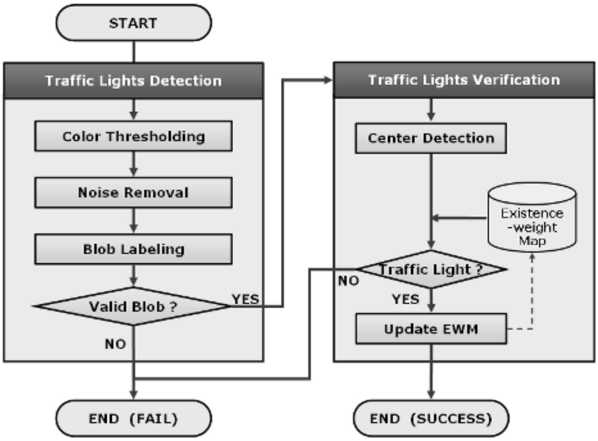


Fig. 1. Traffic lights detection algorithm

$$C(x,y) = \begin{cases} Red & \text{if } (\ ((H(x,y) < T_{rl}) \vee (H(x,y) > T_{ru})) \\ & \qquad \wedge (S(x,y) > T_s) \wedge (I(x,y) > T_i)\ ) \\ Green & \text{if } (\ (T_{gl} < H(x,y) < T_{gu}) \wedge (S(x,y) > T_s) \wedge (I(x,y) > T_i)\ ) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $C(x,y)$ is result of the color thresholding, $H(x,y)$ is Hue value, $S(x,y)$ is saturation value, and $I(x,y)$ is intensity value. The $C(x,y)$ becomes a set of red (include yellow) and green candidate traffic lights. The four threshold values $(T_{rl}, T_{ru}, T_{gl}, T_{gu})$ are determined in order to check the Hue values are included in a predefined range. Further, to detect candidate region of traffic lights whose lights are *on*, we also use thresholds of saturation (T_s) and intensity (T_i).

Noise Removal. There are many noises in the candidate region for traffic lights which come from lens distortion, diffraction of sunlight, vibration of camera, and background objects with similar color. To remove such noises, a morphology operation is executed, as shown in

$$M = ((((C \oplus T) \oplus T) \ominus T) \ominus T) \tag{2}$$

where M is result of morphology operation, C is result of the color thresholding (previous step), T is a 3×3 mask template whose all elements are 1, $\oplus$ means dilation operation, and $\ominus$ means erosion operation. The operation (2) is similar to closing operation of morphology $((C \oplus T) \ominus T)$, but it is a more empirical method that can effectively remove a lot of irregular noises and color components other than those of traffic lights.

Blob Labeling. The segmented candidates of traffic light are processed through blob labeling, and are determined as further candidates if the size of labeled blob meets the criteria

$$N_l < BA < N_u \tag{3}$$

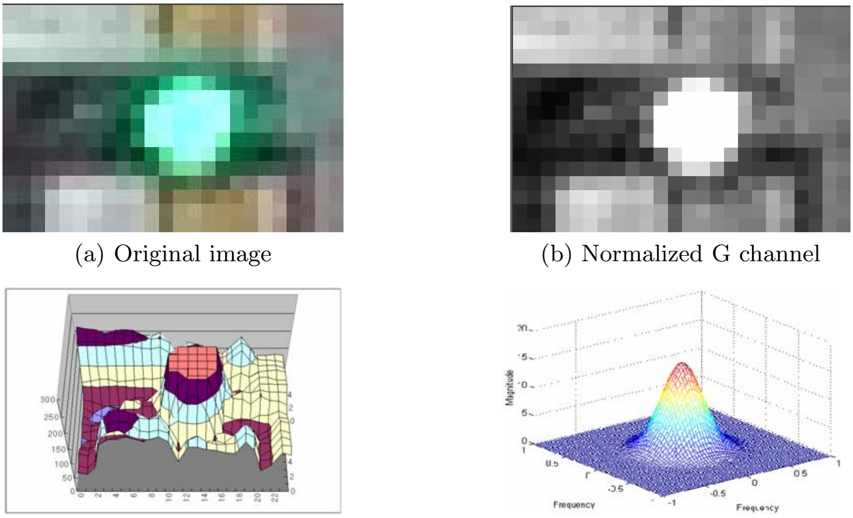
where BA denotes size of blob area, and N_l and N_u are lower bound and upper bound of BA, respectively. That is, only blob whose size is within a certain range is left as a further candidate. The actual size of a traffic light (a lamp) is $30cm$ in radius, while its pixel size depends on the distance between the traffic light and camera, as well as camera specification (such as focal length). The values of N_l and N_u can be dynamically changed, and in our experiments they are set to 30 and 350 respectively.

2.2 Traffic Lights Verification

The detected candidates of traffic lights are processed through the verification process (see algorithm shown in Fig.1). The verification process is executed in two steps described below; finding center of the lights and recognizing the traffic lights using existence-weight map.

Traffic Lights Center Detection. The most useful characteristic of traffic lights is that they emit light themselves. Figure 2 shows the extracted candidates for traffic lights and their normalized distribution with regard to green. We can see the distribution of traffic lights is similar to Gaussian distribution (Fig. 2(c) and (d)). Therefore, we estimate a center of traffic light by Gaussian mask filtering to a given candidate of traffic light. The size of Gaussian mask is set to half of size of the candidate region. Because the size of the candidate region varies (according to the distance between the traffic light and camera), the Gaussian mask should have dynamic size accordingly. We can estimate that a point with largest value of convolution operation on the candidate region and Gaussian mask is center of traffic light. We denote the estimated center of traffic light as (cx, cy). Figure 3(a) shows a candidate of traffic light after labeling process, Fig.3(b) shows a result of estimation of center and boundary of a traffic light, and Fig.3(c) shows the used Gaussian mask.

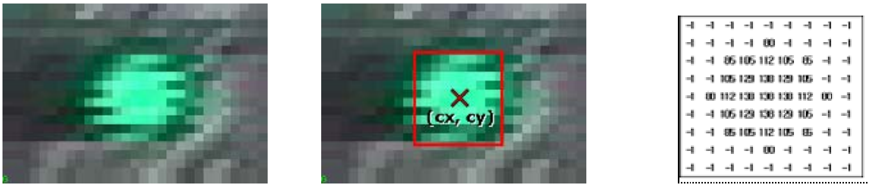
Existence-Weight Map(EWM). In case of moving camera, the detection of traffic lights may fail because of vibration of camera and image distortion caused by light and lens. Especially in case of interlaced camera, the inconsistency of scan line makes the detection more difficult. Considering such difficulties, we use an idea for efficient detection of traffic lights: 'The existence of a traffic light in a frame implies it is very likely that the traffic light also exists at similar pixel position in the next frame.' So we suggest *Existence-Weight Map(EWM)* in order to find a traffic light effectively even in case of uncertain or failed result in a single frame. To define the EWM, we first segment the upper half of image into irregular-sized rectangular grid, which we call *blocks* (Fig.4(a)). We exclude lower half of image because it is nearly impossible for traffic lights to appear, with our camera setting. Note that, the size of each block depends on image size, and it is determined to be larger for upper part of the image because the traffic lights in the image is moving more rapidly around upper part than around center of image.



(a) Original image (b) Normalized G channel

(c) Intensity distribution of (b) (d) Gaussian distribution

Fig. 2. Distribution of traffic lights



-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	00	-1	-1	-1	-1
-1	-1	85	105	112	105	85	-1	-1
-1	-1	105	129	130	129	105	-1	-1
-1	00	112	130	130	130	112	00	-1
-1	-1	105	129	130	129	105	-1	-1
-1	-1	85	105	112	105	85	-1	-1
-1	-1	-1	-1	00	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

(a) Candidate image (b) Detected center (c) Gaussian mask for (b)

Fig. 3. Detecting center of traffic lights

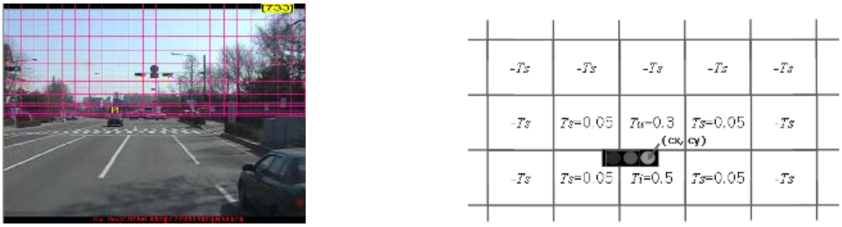
Suppose there are $M \times N$ blocks for each frame t (In our experiments with 720×480 image, we have 24×15 blocks, as shown in Fig.4(a)). For each block a weight value is assigned, and we denote $P(x, y, t)$ as the weight value of (x, y)-th block at t-th frame. Then we define the weight value as *possibility of existence of a traffic light* at each block, and calculate each $P(x, y, t)$ as follows. First, all $P(x, y, t)$ are set to zero before starting to scan the frames. Given frame $t - 1$, let (m, n) be x- and y- index of the block where the estimated center of traffic light (cx, cy) exists. Then each $P(x, y, t)$ at next frame are set to

$$\begin{aligned}
&P(m, n, t) = P(m, n, t-1) + T_i \\
&P(m, n+1, t) = P(m, n+1, t-1) + T_u \\
&P(m \pm 1, n, t) = P(m \pm 1, n, t-1) + T_s \\
&P(m \pm 1, n+1, t) = P(m \pm 1, n+1, t-1) + T_s \\
&P(x, y, t) = P(x, y, t-1) - T_s \\
&\qquad \forall\ x \notin \{m, m-1, m+1\} \ \text{ and } \ \forall\ y \notin \{n, n+1\}
\end{aligned} \tag{4}$$

where the parameters T_i, T_u, T_s are empirically determined positive number with $1 > T_i > T_u > T_s > 0$.

Figure 4(b) shows an example of the parameters values in (4). These values mean: if the center of traffic light is found in block $(m, n, t-1)$, it is most likely to appear in same location (m, n, t), the upward block $(m, n+1, t)$ has the second highest possibility, and four sideward blocks $(m \pm 1, n, t)$ and $(m \pm 1, n+1, t)$ have the third highest possibility. This is because the traffic lights are moving upward or sideward in the next frame as the vehicle is moving. Also, to prevent divergence of $P(x, y, t)$, we limit its value as

$$0 \leq P(x, y, t) \leq 5 \ \ \forall \ x, y, t \qquad (5)$$



(a) Block segmentation (b) Example of parameters

Fig. 4. Existence-weight map(EWM)

Traffic Lights Recognition. With the EWM mentioned above we recognize traffic lights as a final step, because the candidate regions processed by center detection steps may include false candidate as well. We assume that the lamps of traffic lights are distinct from outside region in intensity. To use the intensity criteria, we first divide the candidate region from blob labeling process (see Fig.3) into inner part(lamp part) and outer part(outside estimated boundary of traffic light). Having defined that (m, n) is the index of the block containing the estimated center of traffic light (cx, cy), we recognize the traffic lights using the values of EWM and the intensity values of candidate regions around the estimated traffic light, by the criteria

$$\bar{I}(in) + k \cdot P(m, n, t) > \bar{I}(out) \cdot T_r \qquad (6)$$

where $\bar{I}(in)$ and $\bar{I}(out)$ are average intensity value (defined as 0∼255) of the inner part and the outer part, respectively. In our experiments, the constant k and T_r in (6) is empirically determined as 6 and 2.5 respectively. Finally, we can determine that the (cx, cy) is a center of traffic light if and only if (6) yields true, which is our goal.

3 Interface with Car Navigation System

In this section, we present how the traffic lights detection functions suggested in Section 2 interface and work with car navigation system. The architecture of the navigation system combined with detection module are briefly shown in Fig.5.

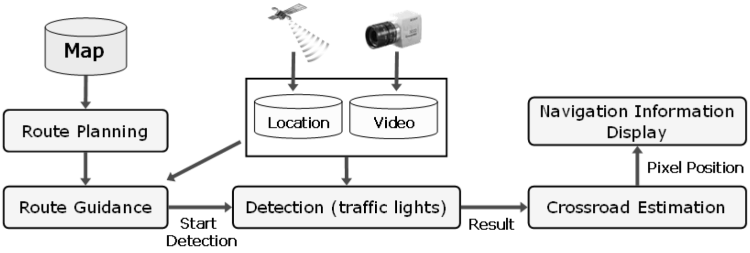


Fig. 5. Architecture of the system

3.1 Sending Notification for Detection

On reaching guide point (crossroad) before a predefined distance, say 300m, the guidance module send a message Detect(current, guide, f, road) to detection module, where current is current position, guide is position of next crossroad, f represents location and attributes of traffic lights, and road is vector data for the road between current and guide. When this message is received, the detection module starts to work with necessary information. The information is mainly used for reducing the search region and estimating where a traffic light appear in a frame. Further, from current and the position of traffic light (included in f) the detection module can estimate how large (pixel size) each traffic light appears in video frame.

3.2 Using Detection Result

After the detection module is executed, the detection results are used to display the navigation information more accurately at crossroad. More specifically, they are used to estimate pixel position that corresponds to real world crossroad. Again, we assume that traffic lights are located at crossroads. For this, the traffic lights located at a crossroad are logically connected to the crossroad in a database. Moreover, for each traffic light, its size (radius), number of lamps, and height from ground are also recorded. The values are basically defined in a domestic transportation standard, so we don't need to measure the values individually. Once a detection result (bounding rectangle of lamp of traffic light) is given, we estimate the pixel position of crossroad using the detection result and the height of the traffic light. Let (x, y) be coordinate of a detection result(bottom of bounding rectangle). Then the estimated pixel position (x', y') of the crossroad is calculated by

$$\begin{aligned} x' &= x \\ y' &= y + (\frac{height \times f}{dist}) \cdot size \end{aligned} \tag{7}$$

where *height* is height of the traffic light from ground, f is focal length of used camera, *dist* is distance between the traffic light and the camera, and *size* is physical length of a CCD cell of the camera (we can get values of f and $size$ from camera specifications). In (7), $y-y'$ means y-axis pixel length corresponding actual length h. After the pixel position that corresponds to real world crossroad is determined, we can get adjusted and more accurate pixel position where a direction indicator should be displayed, even when the simply projected point (camera pitch and topography are not considered) is not accurate.

4 Experiments

The experimental environment for our research is a software application with simulation data. The CPU is Intel Pentium4 processor with 3.0GHz, and the size of main memory is 1024MBytes. The simulation data (video, location, and traffic lights) are collected in two road sections in Seoul and Daejeon, both of which are about $5km$ long. The video is captured with 720×480 size and $30fps$. An application that looks like a video player is developed as a test platform for our experiments.

The detection ratio of the traffic lights detection module is shown in Fig.6. In case of $130m$ or longer distance detection ratio is low because the traffic light appears very small (BA is lower than 10); in case of $90m$ the ratio shows about 80% (BA is about 40). The example detection results with regard to the distance between traffic lights and camera are shown in Fig.7. The average processing time of the traffic lights detection module for each frame is $70.86msec$. Although we do not resize original image size, a processing time of about $15fps$ is possible.

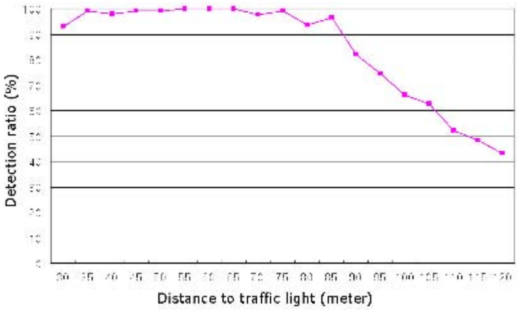
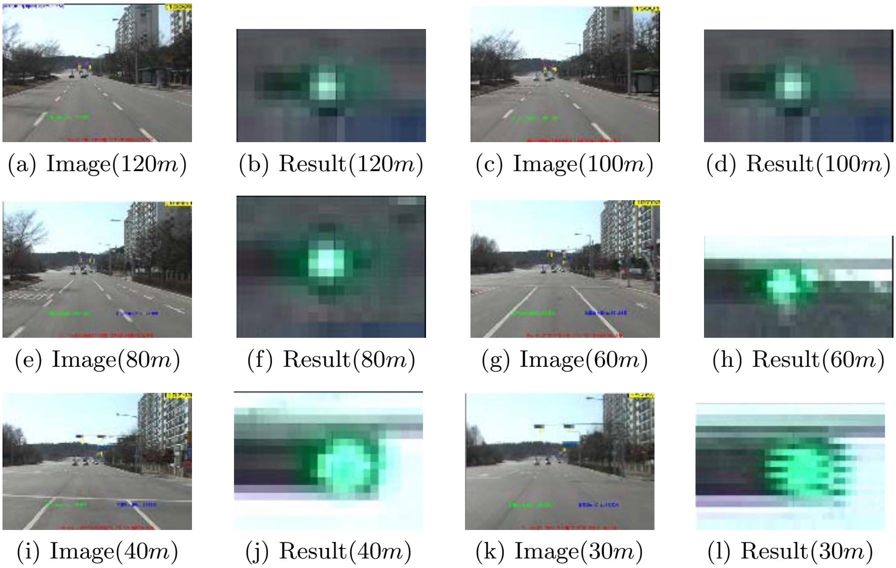


Fig. 6. Detection ratio

With the detected traffic lights and estimated location of crossroad, we display adjusted direction indicator onto video frame at a prototype video-based navigation system. The example of pictorial comparison between simply projected direction indicator and adjusted direction indicator by traffic light detection is shown in Fig.8.

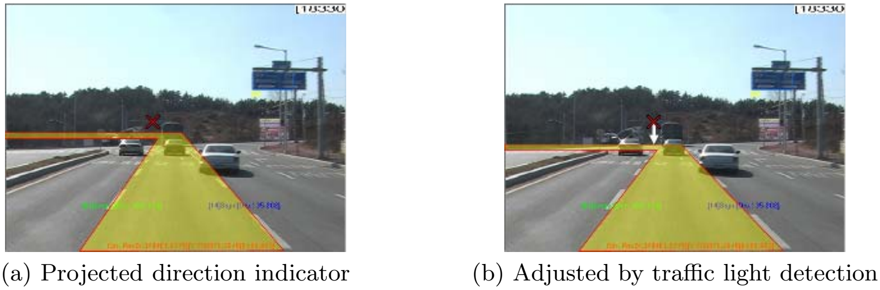


(a) Image(120m) (b) Result(120m) (c) Image(100m) (d) Result(100m)

(e) Image(80m) (f) Result(80m) (g) Image(60m) (h) Result(60m)

(i) Image(40m) (j) Result(40m) (k) Image(30m) (l) Result(30m)

Fig. 7. Detection result



(a) Projected direction indicator (b) Adjusted by traffic light detection

Fig. 8. Adjusted display of direction indicator

5 Conclusion

In this paper, we suggested a detection method for traffic lights that can be effectively used for video-based car navigation systems. For video-based navigation systems that display navigation information overlaid onto video on reaching a crossroad, it is essential to find the exact location of the crossroad in image, because it is important to visually represent guidance information on video.

However, because it is very difficult in actual situation, we suggested a method that effectively detects traffic lights instead of the crossroad itself and estimates location of crossroads with the result. We detect candidate traffic lights by color thresholding, noise removal, and blob labeling process, and then find center of traffic lights by Gaussian mask, which is verified by suggested Existence-Weight

Map. Experiments show that the suggested method works in a long distance before a crossroad for outdoor video and can be effectively used for video-based navigation systems.

Because video is regarded as a new media for future car navigation systems, computer vision technology will be more and more used in car navigation industry. In the future, more computer vision technologies will be required for other objects that appear in video captured by vehicle, such as roadside facilities, buildings, road lanes, or other vehicles. We should develop more robust computer vision methods based on analyses on the demands of applications in the field of car navigation system.

References

1. W. Narzt, G. Pomberger, A. Ferscha, D. Kolb, R. Muller, J. Wieghardt, H. Hortner, C. Lindinger, "Pervasive Information Acquisition for Mobile AR-Navigation Systems," 5th IEEE Workshop on Mobile Computing Systems & Applications, Monterey, California, USA, October 2003, pp.13-20.
2. Zhencheng Hu, Keiichi Uchimura, "Solution of Camera Registration Problem Via 3D-2D Parameterized Model Matching for On-Road Navigation," International Journal of Image and Graphics, Vol.4, No.1, 2004, pp.3-20.
3. Arturo de la Escalera, Jose Maria Armingol, Jose Manuel Pastor, and Francisco Jose Rodriguez, "Visual Sign Information Extraction and Identification by Deformable Models for Intelligent Vehicles," IEEE Transactions on Intelligent Transportation Systems, Vol.5, No.2, Jun 2004.
4. Zhuowen Tu and Ron Li, "Automatic Recognition of Civil Infrastructure Objects in Mobile Mapping Imagery Using Markov Random Field," Proc. of ISPRS Conf. 2000, Amsterdam, Jul 2000.
5. Tae-Hyun Hwang, Seong-Ik Cho, Jong-Hyun Park, and Kyuong-Ho Choi, "Object Tracking for a Video Sequence from a Moving Vehicle: A Multi-modal Approach," ETRI Journal, Vol.28, No.3, 2006, pp.367-370.
6. National Police Agency of Korea, The Standard Guideline for Colors of Traffic Signs, 2004.
7. R. C. Gonzalez and R. C. Woods, 'Digital Image Processing,' Addison-Wesley, 1992.

What Is the Average Human Face?

George Mamic, Clinton Fookes, and Sridha Sridharan

Image and Video Research Laboratory
Queensland University of Technology
GPO Box 2434, 2 George Street,
Brisbane, Queensland 4001
{g.mamic, c.fookes, s.sridharan}@qut.edu.au

Abstract. This paper examines the generation of a generic face model from a moderate sized database. Generic models of a human face have been used in a number of computer vision fields, including reconstruction, active appearance model fitting and face recognition. The model that is constructed in this paper is based upon the mean of the squared errors that are generated by comparing average faces that are calculated from two independent and random samplings of a database of 3D range images. This information is used to determine the average amount of error that is present at given height locations along the generic face based on the number of samples that are considered. These results are then used to sub-region the generic face into areas where the greatest variations occur in the generic face models.

1 Introduction

Today's smart security solutions often employ the use of biometrics, such as fingerprints or iris scans, as an integral part in the overall system. This is largely due to the increased demand on modern society to identify or authenticate an individual with much stronger certainty. The uniqueness of face recognition technology that distinguishes it from the other cohort of biometrics is that images of the face can be captured quickly and in the majority of situations non-intrusively. They also have the potential to operate in noisy crowded environments where other biometrics may fail. Unfortunately, traditional 2D methods for face verification have struggled to cope with changes in illumination and pose. This has led researchers to the use of three-dimensional (3D) facial data in order to overcome these difficulties [1].

One of the key components of 3D face recognition is the ability to reconstruct the 3D human face and estimate relevant models from the captured sensor data. This step is not only important for face recognition, but also for other computer graphics and machine vision fields including entertainment, medical image analysis and human-computer interaction applications [2]. This reconstruction and modeling process often employs the use of a generic face model or generic face mesh as a tool that assists in developing solutions to this challenging problem.

Observation of the numerous and varied approaches to solving the face modeling problem has revealed that little thought is given to the actual generic face

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 692–701, 2006.

model that is employed. Consequently, there are almost as many generic face models that exist in the literature as there are researchers who have investigated the problem. Some face models may be typically constructed from a set of training range images; may use active appearance models [3] or morphable models [4]; or some researchers may even employ the use of their own face as a "generic face model" which is subsequently fitted, deformed, or applied to other people's faces. The latter obviously, is not the most optimum approach to adopt.

This paper attempts to answer the question: what is the average human face and does an average model actually exist? Such questions are rarely asked but the results may be of particular value to a wide range of face modeling and recognition papers. If such a generic/average model does exist, what are the constraints on this model and what are the characteristics of this model? To answer such questions will allow more intelligent application of generic face models in face reconstruction applications.

This paper examines the calculation of the generic face mesh in the following manner. Section 2 presents some existing work in the field and demonstrates how generic face meshes are used by numerous researchers in the face recognition field. Section 3 presents the preprocessing steps that were conducted on the database of 3D images that were used in the construction of the generic face mesh and related experiments. The database employed in this paper is the second version of the "Face Recognition Grand Challenge" (FRGC v2.0) database [5]. Section 4 presents the error metric that was used to determine the number of required models. Section 5 presents the results of the testing of the generic 3D face mesh and this is followed by the Conclusion which is presented in Section 6.

2 Related Work

There are numerous methods to capture 3D information of a human face. Methods include a range of both passive and active sensors such as stereo [6–8], shape from shading [9], structured light [2], shape from silhouettes, laser scanners [5], or structure from motion methods [10]. Some of these methods obtain the 3D information directly while others depend on methods to reliably determine correspondences between images containing human faces. To perform the latter approach in an automatic fashion is extremely problematic. The majority of these problems stem directly from the human face itself, as faces are non-planar, non-rigid, non-lambertian and are subject to self occlusion [11]. Due to these issues and the uniform appearance of large portions of the human face, passive techniques such as stereo can not always accurately determine 3D face shape. Active sensors, although they can produce higher resolution and more accurate data, achieve this with more expensive equipment and at the cost of projecting light or other visible patterns onto the subjects face.

Many passive techniques depend on extensive manual guidance to select specific feature points or determine correspondences between images [12]. Fua and Miccio [8] use a least-squares optimisation to fit a head animation model to stereo data. Although powerful, their method requires the manual selection of key 2D feature points and silhouettes in the stereo images. Ypsilos et al. [2] also fit an animation face model to stereo data. However, the stereo data is combined with a projected infrared structured light pattern allowing the simultaneous capture of 3D shape information as well as colour information. Ansari et al. [13] fit a face model to stereo data via a Procrustes analysis to globally optimise the fit, then a local deformation is performed to refine the fitting. Some methods use optical flow [14] or appearance-based techniques [10] to overcome the lack of texture in many regions in the face. These approaches, however, are often hampered by small deformations between successive images and issues associated with non-uniform illumination.

Differential techniques attempt to characterise the salient facial features such as the nose, the mouth and the orbits using extracted features such as crest lines and curvature information. Lengagne et al. [15] employ a constrained mesh optimisation based on energy minimisation. The energy function is a combination of two terms, an external term, which fits the model to the data and an internal term, which is a regularisation term. The difficulty with differential techniques is in controlling the optimisation as noisy images can wreak havoc with estimating differential quantities, whilst regularisation terms have the potential to smooth out the very features that are most important to recognition algorithms.

Statistical techniques fit morphable models of 3D faces to images using information that is gathered during the construction of the generic face mesh. Blanz and Vetter [16] present a method for face recognition, by simulating the process of image formation in 3D space, fitting a morphable model of 3D faces to images and a framework for face identification based upon model parameters defined by 3D shape and texture. The developed system is a complete image analysis system provided that manual interaction in the definition of feature points is permissible. Koterba et al. [3] however, studied the relationship between active appearance model fitting and camera calibration. They use a generic model as a non-rigid calibration grid to improve the performance of multi-view active appearance model fitting.

3 Face Mesh Preprocessing

The first stage in the investigation of the construction of the generic face mesh involved the development of an appropriate database of face images that could be used. Sets of range images were taken from the FRGC v2.0 database [5] and each of the images was then subjected to three stage pre-processing. This data normalisation stage is an integral step for any face recognition application, particularly for 3D data. It is rarely noted but the 3D modality has high levels of noise with often extreme outliers that must consequently be removed. Another issue rarely raised with 3D face recognition is there is a pose dependence. The angle of the individual to the laser scanner (or other capture device) will cause self-occlusion, unless the capture equipment can gather a full 3D face image. However, even with these issues 3D face data is considered to be less susceptible to noise than 2D images.

The first stage of the pre-processing pose-normalised the data such that the position and direction of the normal vector of the plane formed by the centre of the eyes and the tip of the nose was identical across all the images. The second stage filtered and interpolated across areas of the face that contained substantial errors in the range image, as is caused when taking range measurements in the eyeball area and the nostrils. The final stage of the pre-processing cropped all the images so as to only contain information from the forehead of the face down to the top lip of the subjects. This was done primarily to reduce any errors that may be introduced by variations in the mouth region that may have occurred during the image capture process. These stages are outlined in more detail as follows [17].

The 3D data is assumed to be point cloud data on a semi-regular x- and y-grid; there is limited variation along the grid lines. Every image has four landmark points: the right eye corner, left eye corner, nose tip and chin. Depth normalisation is conducted by using three of these landmark points (right eye corner, left eye corner and chin) to compensate for in-plane and out-of-plane rotations; the three landmark points are rotated to reside in the plane. The face data is then cropped based on the two eye corners and the erroneous data in this cropped region is removed using a gradient filter (Algorithm 1) in the x and y domains respectively, $Z_{norm1} = grad\_filt(Z, \partial x, N)$ and $Z_{norm2} = grad\_filt(Z_{norm1}, \partial y, N)$; the median filter used is defined by Algorithm 2. Where Z is the cropped depth image, N is the size of the median filter, C is the cutoff point for the gradient filter, i refers to the columns in the image and j refers to the rows in the image.

Algorithm 1. Gradient Filter(grad_filt($Z, \partial n, N$))

```
$Z_n = \frac{\partial Z}{\partial n}$
**if** $Z_n(i,j) > C$ **then**
    $Z(i,j) = error$
**end if**
**for** $i = 1 : \text{num\_cols}(Z)$ **do**
    **for** $j = 1 : \text{num\_rows}(Z)$ **do**
        $Z(i,j) = \text{med\_filt}(Z, i, j, N)$
    **end for**
**end for**
```

For this work it was found that a value of $C = 10$ was appropriate and that a median filter of size $N = 5$ was sufficient; for images which were larger than 128×128 pixels. This valid data, Z_{norm2} was re-interpolated onto a regular grid of 128×128 pixels, an example output of this procedure is provided in Figure 1.

In addition to filtering and registration, the 3D data must also be range normalised. Range normalisation consists of setting the maximum value to 255 (the maximum value for a normal 2D image) and adjusting all other values to be relative to this value.

Algorithm 2. Median Filter (med_filt(Z, i, j, N))

```
start_i $= i - N$, stop_i $= i + N$
start_j $= j - N$, stop_j $= j + N$
**if** start_i $< 1$ **then**
    start_i $= 1$
**else if** stop_i $>$ num_columns$(Z)$ **then**
    stop_i $=$ num_columns$(Z)$
**end if**
**if** start_j $< 1$ **then**
    start_j $= 1$
**else if** stop_j $>$ num_rows$(Z)$ **then**
    stop_j $=$ num_rows$(Z)$
**end if**
$Z_{sub\_region} = Z(\text{start\_i} : \text{stop\_i}, \text{start\_j} : \text{stop\_j})$
$Z(i, j) = median(Z_{sub\_region} \neq \text{error})$
```

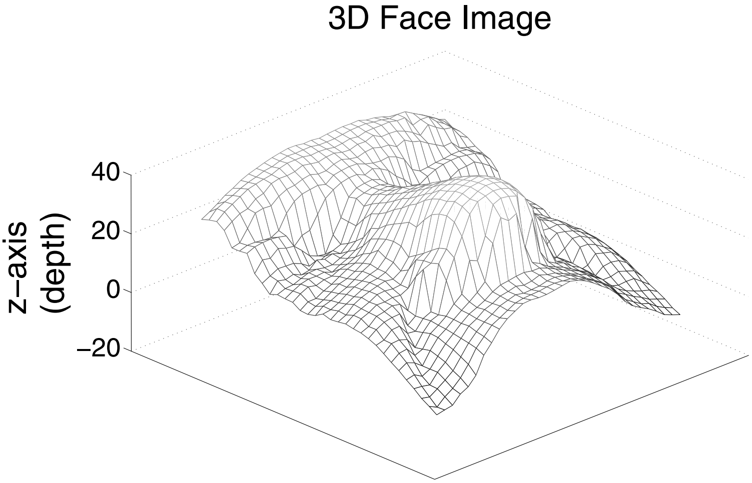


Fig. 1. A mesh plot of a cropped and interpolated 3D face image

The final database was composed of 466 individuals represented by 128×128 images. Each image has the spatial co-ordinates available in X, Y and Z matrices and the position of the eye centres and nose tip was available in a separate structure.

4 Generic Face Experiment Design

The investigation of the average face was done by drawing two random and independent samples of the database of range images R that was constructed in the preprocessing stage. The random sampling was performed using a random number generator which generates uniformly distributed pseudo-random numbers. The state of the random number generator was altered with every iteration.

For a given set, S, of n randomly selected range images, where $S \subset R$, the mean range distances, $M1$, were calculated by,

$$M1_{X,Y} = \frac{\sum_{i=1}^{n} Si_{X,Y}}{n}. \tag{1}$$

A second set T, of n random range images, where $T \subset Q$ and $Q = R - S$, was then chosen and the mean range distances $M2$ of this set of images was calculated in a similar manner.

To compare the two 'average' faces that were calculated from set S and set T a squared error matrix E^2 was calculated using,

$$E^2 = [M_1 - M_2]^2. \tag{2}$$

The final matrix E^2 provides the squared error that exists at each point on the surface for the particular sets of randomly generated mean images.

This procedure for generating the matrix E^2 was then repeated for 10,000 simulation runs. The collection of the squared error matrices were then used to determine the average squared error across all simulation runs in the following manner,

$$E^2_{ave} = E^2/10000. \tag{3}$$

5 Results and Discussion

To investigate the MSE versus the number of models that were present in the database of objects, the experiment detailed in the previous section was repeated for $n = 1...233$ (i.e. half the total database size of 466 due to the creation of two generic face models being compared). For every value of n the experiment was conducted 10000 times and the MSE at each observed range measurement was calculated. The results of this set of experiments are presented in Figure 2.

Figure 2 shows that there is an inverse relationship between the number of models and the amount of mean squared error which is present between the mean squared faces that are calculated. As this is an inverse relationship, it indicates that a truly average face would be achieved only with an infinitely large database. However, by examining the mean error that exists at any given location between the average faces that are calculated for a given sample size it is possible to calculate the percentage mean squared error that can be expected to be found at given locations across an average face. These results are presented in Table 1.

As seen from Table 1, once more than sixty models are chosen to calculate the average face, the average squared error at a given location on the face varies by 5% or less to another average face which is calculated using the same number of models. This result should serve as a guide as to how many faces should be 'averaged' when calculating a generic face to represent the faces that are present

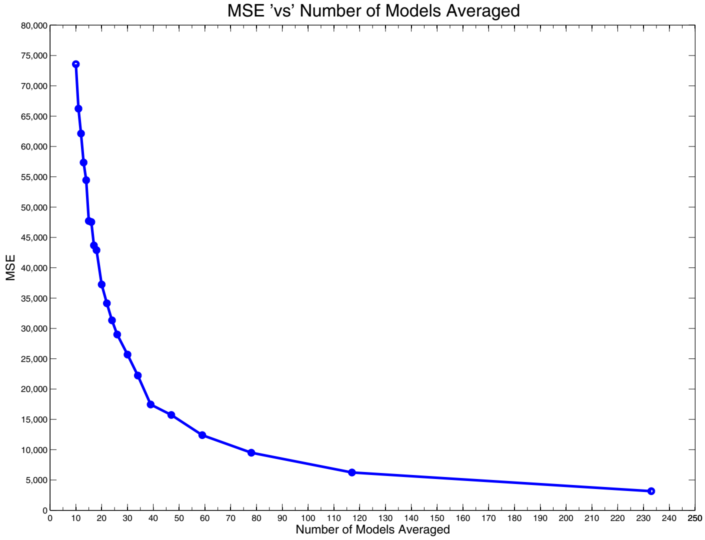


Fig. 2. Plot of MSE versus the number of faces in the model averaged

in a database. This can also be used as a starting point in reconstruction and calibration algorithms which require a generic model for initialisation of the algorithm.

To further investigate the relationship that exists between the MSE and the 'average face', the MSE surface as it exists on the average face was plotted for the case where 233 models were considered. This relationship is displayed in Figure 3.

Figure 3 shows that the bulk of the MSE that exists between the average models was located at the tip of the nasal region and around the brow area, where as the cheeks and forehead region on the other hand had far lower values of MSE associated with them. This result indicates that the most important information for applications such as recognition and face reconstruction occurs in these areas, and thus algorithms which are time critical in their application should focus on these regions for salient parameter calculations. This result is also important for any coding standards that are produced for faces, as any meshes used to represent the face should focus the vertices in the nasal and brow areas. To further illustrate this point, a face that contains 95% of the MSE that exists when 233 models are considered can be represented as shown in Figure 4.

The results that have been presented in this paper have been empirical in nature. A formal hypothesis testing procedure was constructed using the Bonferroni correction as the mechanism by which every range point on the image could be tested simultaneously. Although this framework was accurate in a synthetic

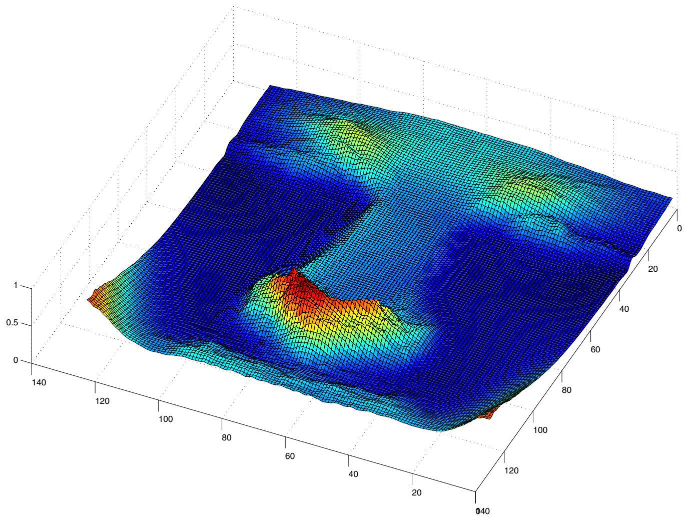


Fig. 3. Plot of MSE variations over the generic face model surface

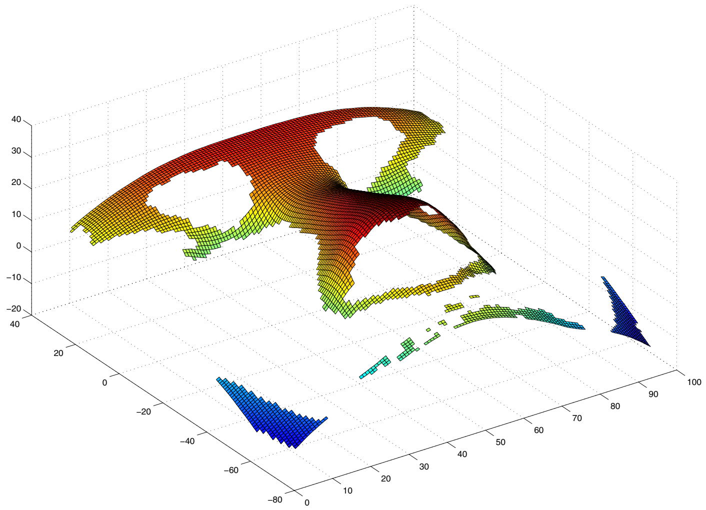


Fig. 4. Plot of face regions that contain 95% of the MSE that exists in the generic face model when 233 faces are utilised

environment, the variations and interpolation errors for the eye and nostril regions that are present on a database of real faces provided inconsistent results. This is currently the focus of future research and it is hoped that a formal framework will eventually be able to be constructed for facial analysis.

Table 1. Table of percentage MSE versus the number of models

No. Models	% Error
10	31.71%
15	20.57%
20	16.05%
24	13.50%
30	11.07%
34	9.58%
39	7.52%
47	6.78%
59	5.34%
78	4.10%
117	2.69%
233	1.36%

6 Conclusion

This paper examined the generation of a generic face model from a moderate sized database. The model constructed was based upon the mean of the squared errors that are generated by comparing average faces that are calculated from two independent and random samplings of a database of 3D range images. This information is used to determine the average amount of error that is present at given height locations along the generic face based on the number of samples that are considered. Experimental results demonstrated that at least sixty faces need to be incorporated into a generic face model in order to obtain a mean squared error of 5% or less to another average face which is calculated using the same number of models. The empirical evidence presented has demonstrated that a generic or 'average' face model does exist down to an acceptable level of error. The constructed model also revealed that 95% of the errors within the average model are contained within the brow and nose regions of the face. The presented empirical investigation of the generic face model will allow for more intelligent application of generic face models in face reconstruction and recognition applications.

References

1. Bowyer, K.C.K.W., Flynn, P.J.: A survey of approaches to three-dimensional face recognition. Technical report, University of Notre Dame (2003)
2. Ypsilos, I., Hilton, A., Rowe, S.: Video-rate capture of dynamic face shape and appearance. In: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition. (2004)
3. Koterba, S., Baker, S., Matthews, I., Changbo, H., Xiao, J., Cohn, J., Kanade, T.: Multi-view aam fitting and camera calibration. In: Tenth IEEE International Conference on Computer Vision. (2005) 511–518
4. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH. (1999)

5. Phillips, J., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: Proceedings of IEEE Conference of Computer Vision and Pattern Recognition. Volume 1. (2005) 947–954
6. Kuo, C., Lin, T.G., Huang, R.S., Odeh, S.: Facial model estimation from stereo/mono image sequence. IEEE Transactions on Multimedia **5**(1) (2003) 8–23
7. Chen, Q., Medioni, G.: Building 3d human face models from two photographs. Journal of VLSI Signal Processing **27** (2001) 127–140
8. Fua, P., Miccio, C.: Animated heads from ordinary images: A least-squares approach. Computer Vision and Image Understanding **75**(3) (1999) 247–259
9. Choi, K., Worthington, P., Hancock, E.: Estimating facial pose using shape-from-shading. Pattern Recognition Letters **23**(5) (2002) 533–548
10. Kang, S.: A structure from motion approach using constrained deformable models and apperance prediction. Technical Report CRL 97/6, Cambridge Research Laboratory (1997)
11. Baker, S., Kanade, T.: Super-resolution optical flow. Technical Report CMU-RI-TR-99-36, Robotics Institute, Carnegie Mellon University (1999)
12. Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., Salesin, D.: Synthesizing realistic facial expressions from photographs. In: Proceedings of SIGGRAPH Computer Graphics. Volume 26. (1998) 75–84
13. Ansari, A.N., Abdel-Mottaleb, M.: 3d face modelling using two views and a generic face model with application to 3d face recognition. In: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance. (2003)
14. DeCarlo, D., Metaxas, D.: Deformable model-based shape and motion analysis from images using motion residual error. In: International Conference on Computer Vision, India (1998) 113–119
15. Lengagne, R., Fua, P., Monga, O.: 3d stereo reconstruction of human faces driven by differential constraints. Image and Vision Computing **18** (2000) 337–343
16. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(9) (2003) 1063–1074
17. McCool, C., Chandran, V., Sridharan, S.: 2d-3d hybrid face recognition based on pca and feature modelling. Proceedings of the 2nd Workshop on Multimodal User Authentication (2006)

An Improved Real-Time Contour Tracking Algorithm Using Fast Level Set Method

Myo Thida[1], Kap Luk Chan[2], and How-Lung Eng[3]

[1,2]Center for Signal Processing, Division of Information Engineering,
School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore, 639798,
[3] Institute for Infocomm Research, 21 Heng Mui Keng Terrace Singapore 119613
[1]myot0004@pmail.ntu.edu.sg,[2]eklchan@ntu.edu.sg,[3]hleng@i2r.a-star.edu.sg

Abstract. Human contour provides important information in high level vision tasks such as activity recognition and human computer interaction where detailed analysis of shape deformation is required. In this paper, a real time region-based contour tracking algorithm is presented. The main advantage of our algorithm is the ability of tracking nonrigid objects such as human using an adaptive external speed function with fast level set method. The weighting parameter, λ_i is adjusted to accommodate the situations in which the object being tracked and the background have similar intensity. Experimental results show the better performance of the proposed algorithm with adaptive weights.

1 Introduction

Successful human contour tracking plays a key role in identifying people and recognizing or understanding human behaviors. In early years, human tracking concentrates on feature points such as corner points of moving silhouettes or centroid of rectangular box that bounded the tracked person. However, feature-based method is not appropriate when the shape of the objects changes dynamically as the shape of objects cannot be easily related to the positions of feature points in subsequent frames. In addition, human contour provides important information in high level vision tasks such as activity recognition and human computer interaction where detailed analysis of the shape deformation is required.

The active contour model, also known as the Snake model, was proposed by Kass et al.[7] in 1987 and it had been widely used in edge detection, shape modeling, segmentation and motion tracking. However, traditional active contour approach encounters two major problems: the mathematical model in the snake method cannot deal with topological change without adding external mechanisms, and it is very sensitive to initial contour position and initial shape.

The Level Set Method introduced by Osher and Sethian [12] overcomes the above problems. Using the level set formulation, complex curves can be detected and tracked and topological change of the curves are naturally managed. Considerable research has been reported on applying the Level Set Method in moving

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 702–711, 2006.

object detection and tracking. Paragios and Deriche[10] proposed level-set based geodesic framework for object tracking where speed function in the formulation is defined in terms of image gradient, ∇I. But, tracking using gradient image will fail when the boundary between background image and object being tracked becomes ambiguous. By generalizing the region competition idea [13], Mansouri [8] proposed Bayesian-based level set tracking method where the object being tracked and background regions are assumed to have distinct luminance characteristics. In [9], Paragios and Deriche combine region-based and edge-based information. Recently, many researchers have introduced shape information into level set method [1], [5]. However, the use of Level Set Method has been limited due to its high computational complexity. The major challenges of using level set method in object tracking are:(1) reducing computational cost for implementing and (2)constructing an adequate speed function that governs the curve deformation. In [11], a new implementation of level set method is proposed and achieved real time performance. In this paper, a novel method for human contour tracking using the fast level set implementation method is proposed where the speed function is defined based on the Chan-Vese Model with adaptive weighting parameters, λ_i. Compared to [11] which mainly focus on implementation of level set method in real time, our main objective is to construct an adequate speed function that is adaptable to fast level set implementation so that our tracking algorithm is robust to weak object boundary while achieving real time performance.

In the next section, we briefly review the level set method and discuss the fast level set implementation in [11]. Section 3 presents the proposed algorithm and explains the formulation of external speed function in detail. Finally, experimental results and concluding remarks are given in Sections 4 and 5.

2 Level Set Method

The level set method is a numerical technique for tracking a propagating interface which changes topology over time. This method has been widely used in many areas including computer vision and image proposing after the first work was introduced by Osher and Sethian [12]. The key attraction of the level set method over the snake method is its ability to handle changes in curve topology in a natural way. In addition, the level set method is not sensitive to initial contour position and shape as snake method do. This makes the level set method more suitable for detecting and tracking of multiple objects simultaneously while handling topological changes in natural way.

Level set method embeds the initial position of moving interface as the zero level set of higher dimensional function $\phi(x, y, t)$ [12]. Then, the evolution of implicit function is tracked over time. Finally, the position of the moving interface at anytime, t, is given by the zero level set of time-dependent level set function. That is,$C(x, y, t) = \{(x, y)|\phi(x, y, t) = 0\}$. As the interface moves over time, the value of the function ϕ at each grid point (x, y) must be updated according to the speed function F. The speed function F can be generated based on global properties of the image or local properties such as curvature. Regardless of the properties that generate the speed function, it can be assumed that the speed function is in the direction perpendicular to the interface as the tangential component will not effect on the position of the interface. In order to represent the evolving interface as the zero level set of the implicit function, ϕ ,the following condition must be true at any time t:

$$\phi(p(x(t), y(t)), t) = 0 \tag{1}$$

By taking derivative using the chain rule, we have

$$\phi_t + \nabla\phi(p(t), t).p'(t) = 0 \tag{2}$$

Let $\overrightarrow{n}$ be a local unit(outward) normal to the interface, and then, $\overrightarrow{n}$ can be written in terms of ϕ as follows:

$$\overrightarrow{n} = \frac{\nabla\phi}{|\nabla\phi|} \tag{3}$$

and speed F which is perpendicular to the interface is

$$F = p'(t)\overrightarrow{n} \tag{4}$$

Hence, equation (2) becomes

$$\phi_t + F|\nabla\phi| = 0, \quad \phi(x, y, 0) = \phi_0(x, y) \tag{5}$$

where the set $\{(x, y)|\phi_0(x, y) = 0\}$ defines the initial contour.

2.1 Speed Function, *F*

The performance of level set function depends on the speed function F that governs the model deformation. In classical approach, level set speed function depends on the image gradient, ∇I. For example, Geodesic Active Contours [10] sets the speed function F as:

$$F = g(|\nabla I|)(\kappa + \nu) \tag{6}$$

where $g(|\nabla I|)$ is commonly defined as

$$g(|\nabla I|) = \frac{1}{1 + |\nabla G_\sigma * I|^p} \tag{7}$$

with $p = 1$ or $p = 2$. G_σ denotes a Gaussian convolution filter with standard deviation σ. The speed decreases towards zero at high gradient location. The main assumption for this edge-based model is that the object being tracked has strong intensity contrast to the background. This assumption constraints the applications of the gradient based level set method. This had led researchers to consider the region-based energy which rely on the information provided by the entire region such as color, texture and motion-based properties [2]. Motivated by [13], the speed function that considers the probability density function of the regions inside and outside the structures of interest have been proposed [1], [8]. This class of speed function is well suited in the situations where intensity distributions of regions match to the assumed intensity models.

2.2 Implementation Technique

In the traditional approach, the level set equation in (5) is solved by replacing time derivatives with finite differences and spatial derivatives with approximations using forward and backward differences. But, the direct implementation approach of level set method is computationally very expensive and the Narrow Band approach has been recently popular where the computation is limited to only the neighborhood pixels of the zero level set[1]. However, the computational cost still remains considerable and real time performance is not achieved.

Shi et al. [11] proposed a fast implementation technique for solving level set method. Their approach differs from the Narrow Band Method that in their method, the curve is evolved without solving partial different equation though the evolving curve is still represented by level set function ϕ. The computational cost drops dramatically and real time performance can be achieved.

Fast Level Set Implementation Approach. In the level set method, the boundary curve C is represented implicitly as the zero level set of a function ϕ. In classical level set method, the moving object is tracked by solving the following partial differential equation.

$$\phi_t + F|\nabla\phi| = 0 \tag{8}$$

Given the initial contour position, the value for $\phi(x, y, t = 0)$ is initialized. Generally, ϕ is defined as signed distance function which is negative inside the curve C and positive outside C. Then, the value of ϕ is adjusted based on the speed function F that specifies how contour points move in time.

By examining the neighbor points along the contour, it can be seen that the values of ϕ of neighboring grid point inside C will be switched from negative to positive if the curve moves inward and vice versa if curve moves outward. Based on the above observations, fast level set method defines two lists of neighboring grid points L_{in} and L_{out} for the boundary curve C as follows:

$$\begin{aligned} L_{in} &= \{x|\phi(x) < 0 \text{ and } \exists y \in N_4(x) \text{ such that } \phi(y) > 0\} \\ L_{out} &= \{x|\phi(x) > 0 \text{ and } \exists y \in N_4(x) \text{ such that } \phi(y) < 0\} \end{aligned}$$

where $N_4(x)$ is 4-connected discrete neighborhood of a pixel x. The value of ϕ is chosen from a limited set of integers $\{-3, -1, 1, 3\}$ and defined as:

$$\phi(x) = \begin{cases} 3, \text{if x is exterior point;} \\ 1, \text{if x is in } L_{out}; \\ -1, \text{if x is in } L_{in}; \\ -3, \text{if x is interior point.} \end{cases} \tag{9}$$

where the exterior points are those pixels outside C but not in L_{out} and interior points are those pixels inside C but not in L_{in}. Then, the curve C is evolved at pixel resolution by simply switching the neighboring pixels between the two lists

L_{in} and L_{out} based on the speed function F. In General, the speed function F is composed of external and internal image dependent speed:

$$\frac{\partial \phi}{\partial t} = (F_{ext} + F_{int})|\nabla \phi| \tag{10}$$

The evolution due to data dependent speed F_{ext} and the smoothness regularization due to F_{int} is separated into two cycles. The first cycle moves the curve toward the desired boundary based on the external speed, F_{ext} and the second cycle smooths the curve with F_{int}. In general, F_{ext} is data dependent speed and defined as

$$F_{ext} = \begin{cases} > 0, \text{ if a pixel lies inside the object;} \\ < 0, \text{ otherwise.} \end{cases}$$

According to the definition of L_{in} and L_{out}, the external speed values should be positive for all the pixels in L_{in} and negative for the pixels in L_{out} if the curve is at the object boundary. Based on this observation, we switch a pixel from L_{out} to L_{in} if $F > 0$ and vice versa if $F < 0$. The iteration process stops (a) if the speed at each neighboring grid points satisfy:

$$\begin{aligned} F(x) \leq 0 \quad \forall x \in L_{out} \\ F(x) \geq 0 \quad \forall x \in L_{in} \end{aligned}$$

or (b) A pre-specified maximum number of iterations, N_a is reached. A predefined maximum iteration number, N_a is required for the image with noises and clutter conditions. The popular choice for F_{int} is $\mu\kappa$ where κ is curvature of the curve and defined as the Laplacian of ϕ.

3 Proposed Algorithm

Assuming that the image acquisition system is still, the moving object is detected by gray-level differencing between the current frame and reference background model. The background model is updated with time using a mixture of K Gaussian distributions [3]. Ideally, the difference image is composed of two regions: static and moving object. The static region corresponds to those pixels that belong to both current and background frame and differences are approximately zero. The moving region correspond to those pixels that belong to only the current frame and differences are significantly large. This property can be exploited to define the region-based speed function. Difference from [11], we define the region-based speed function based on the Chan-Vese Model for region segmentation [4].

3.1 Speed Function Based on Region Intensity

Given the difference image $I(x, y) = I_t(x, y) - B(x, y)$ where $I_t(x, y)$ and $B(x, y)$ refers to the current image and background image respectively, the energy function of Chan-Vese segmentation model [4] can be written as

$$E = \mu.|\partial\Omega_{in}|+v.\int_{\Omega_{in}} dx+\lambda_1\int_{\Omega_{in}}(I(x,y)-c_1)^2dxdy+\lambda_2\int_{\Omega_{out}}(I(x,y)-c_2)^2dxdy$$

where Ω_{in} and Ω_{out} denotes the object and background region respectively.

The term $|\partial\Omega_{in}|$ is the length of the object boundary and it provides smoothness regularization. The second term penalizes the area of the object region and is generally omitted by taking v to be zero. The last two terms represents the region fitting energy and will be minimized only if the moving curve is at the boundary of the object being tracked.

The parameters c_1 and c_2 represents the averages of image inside and outside curve C and are updated via the following equation as ϕ evolves:

$$c_1(\phi) = \frac{\int_\Omega I(x,y)H(\phi)dxdy}{\int_\Omega H(\phi)dxdy}; \quad c_2(\phi) = \frac{\int_\Omega I(x,y)(1-H(\phi))dxdy}{\int_\Omega(1-H(\phi))dxdy}$$

where $H(\phi)$ is Heavisdie function and defined as

$$H(x) = \begin{cases} 1, x < 0; \\ 0, x > 0. \end{cases}$$

Keeping c_1 and c_2 fixed and minimizing the energy function, E, with respect to ϕ results

$$\frac{\partial\phi}{\partial t} = \delta(\phi)\left[\mu\nabla.(\frac{\nabla\phi}{|\nabla\phi|}) - v - \lambda_1(I(x,y)-c_1)^2 + \lambda_2(I(x,y)-c_2)^2\right] \qquad (11)$$

In order to extend the evolution to all level sets of ϕ , Dirac function $\delta(\phi)$ can be replaced with $|\nabla\phi|$. Hence, above equation becomes

$$\frac{\partial\phi}{\partial t} = |\nabla(\phi)|\left[\mu\nabla.(\frac{\nabla\phi}{|\nabla\phi|}) - v - \lambda_1(I(x,y)-c_1)^2 + \lambda_2(I(x,y)-c_2)^2\right] \qquad (12)$$

where $\mu > 0, v \geq 0$, and $\lambda_1, \lambda_2 > 0$ are fixed parameters. In fast level set implementation, the level set equation is of the form

$$\frac{\partial\phi}{\partial t} = (F_{ext} + F_{int})|\nabla\phi| \qquad (13)$$

where F_{ext} drives the curve towards the desired boundary and F_{int} smooths the curve. To provide such behavior, we define the speed function as:

$$F_{ext} = \text{sign}(-\lambda_1(I(x,y)-c_1)^2 + \lambda_2(I(x,y)-c_2)^2) \qquad (14)$$

$$F_{int} = \mu\nabla.(\frac{\nabla\phi}{|\nabla\phi|}) \qquad (15)$$

In equation (14), $I(x,y) \approx c_1$ for a pixel inside the object and $I(x,y) \approx c_2$ for pixels outside the object. Hence, the first term in F_{ext} will be much smaller than the second term and the resultant F_{ext} will be positive if a pixel lies inside the moving object. On the other hand, the external speed F_{ext} will be negative if the pixel lies outside the moving object. The negative and positive values of F_{ext} drives the curve toward the object boundary where $F_{ext} = 0$. Internal speed, F_{int} comes from the length constraints and provides the boundary smoothness and length scale of the moving object.

3.2 Curve Evolution Parameters

The parameters, λ_i gives the weights to the object and background classes. If we choose weighting parameters, λ_i to be the same, it implies that both object and background classes are equally likely to occur and no particular class is favored over another. In our proposed algorithm, we adaptively define the external speed function to reduce the misclassification of foreground pixels as background through the parameter λ_i. Let $\lambda_2 = k\lambda_1$ Then, the external speed is redefined as:

$$F_{ext} = \begin{cases} -\lambda_1(I(x,y) - c_1)^2 + k\lambda_1(I(x,y) - c_2)^2, \text{ if } |F_{ext}| < th; \\ -(I(x,y) - c_1)^2 + (I(x,y) - c_2)^2, \qquad \text{otherwise} \end{cases} \tag{16}$$

where k is a constant parameter and $k \geq 1$. The parameter, th is a threshold determining the weakness of external speed function. In our algorithm, the parameters, c_1 and c_2 are chosen to represent the mean intensity values and can extend our algorithm by choosing parameters, c_1 and c_2 to be other statistical properties.

4 Experimental Results

To evaluate the performance of proposed method, it is tested with real time video sequences. The first few frame is assumed to have no moving objects and assigned as the reference background model. For the subsequent frames, the difference image between the current frame and background model is used as input to the tracking algorithm while background model is updated with time.

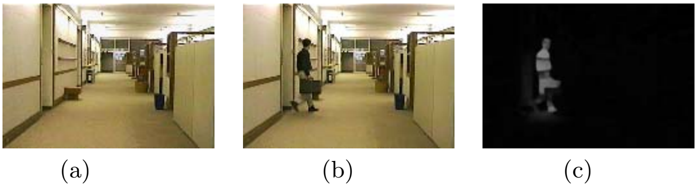
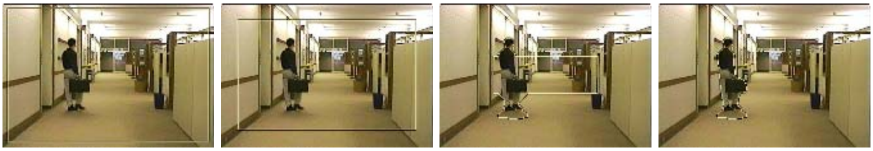

(a) (b) (c)

Fig. 1. (a) Background Model (b) Current Frame and (c) Difference Image

According to the fast level set implementation [11], the mathematical implementation is performed in two cycle. The regularization parameters, N_a and N_g are selected based on the balance between the external speed and the smoothing effects. In the proposed algorithm, the initial curve can be anywhere in the image and we set the image border as the initial curve in order to detect the moving object in the whole domain. For the subsequent frames, the initial curve can be obtained by the results from the previous frame. In our experiment, we choose $N_a = 255$ for motion detection in the first frame and 20 for subsequent frames. By setting the parameter, N_a selectively, we can further reduce the computational cost. The parameter, N_g eliminates the small holes in the final result and is chosen based on the noise level of the input image. Generally, N_g is chosen to be much smaller than N_a. In our experiment, we set $N_g = 7$ and the parameter, N_g is fixed for the whole sequence.

4.1 Motion Detection

In the proposed algorithm, the initial curve can be anywhere in the image and we set the image border as the initial curve in order to detect the moving objects in the whole image. The curve will be initialized to the image border whenever the motion detection module shows there are changes in the number of moving objects in the scene. Otherwise, the tracking curve in the previous frame will be used to initialize the curve in the current frame.Fig. (2) shows the evolution process in one of the frames of *Hall monitor* sequence. The contour of the person is successfully detected after 243 iterations by satisfying part (a) of stopping conditions. The time for one iteration is about 0.005 sec.

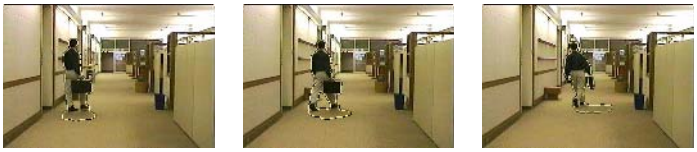


iteration_11 iteration_51 iteration_171 iteration_243

Fig. 2. The curve evolution process of the two cycle algorithm

4.2 Tracking

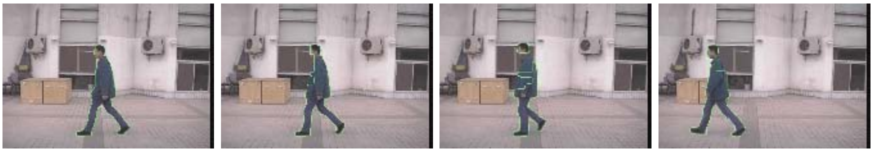
After the moving object is detected successfully, only a few number of iterations are required in subsequent frames to obtain tight boundary. The average number of iterations required in the subsequent frames is 8 and average tracking time is 0.0866 sec per frame on a 2.4 GHz PC. The moving person and his shadow is tracked successfully in real time as he walks in the hallway.



frame 30 frame 35 frame 50

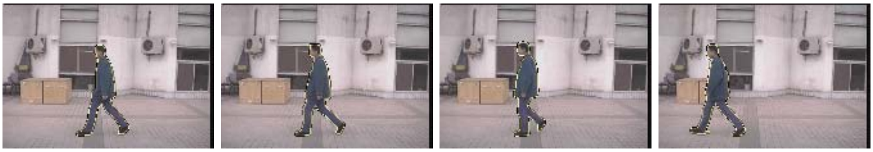
Fig. 3. Tracking results of *Hall monitor* sequence. Frame 30, 35, and 50 are shown.

We present the result of applying our adaptive speed function to one of the sequences from NLPR database [6]. The boundaries of human silhouettes in the observed images are tracked using our proposed algorithm. In the first



frame 23 frame 24 frame 27 frame 36

Fig. 4. Tracking results of *the moving person.* Frame 22, 23 and 24 are shown. Equal weighting parameters: $\lambda_1 = \lambda_2 = 1$.



frame 23 frame 24 frame 27 frame 36

Fig. 5. Tracking results of *the moving person*: Giving object class a higher weight for those pixels with $|F_{ext}| < th$

frame, the curves converge to the boundary of the person to be tracked after about 239 iterations.

From frame 23 to frame 36, the intensity values of the background and the part of the tracked person's body are similar. The frame difference for most of the points in the left side of body part is very small and misled the algorithm to define these points as background points Fig.(4). Indeed, this is a well-known and challenging problem in region-based level set algorithm. In the proposed algorithm as well as other region-based level set method, tracking is treated as discriminant analysis of pixels into two classes, where the classes correspond to object region and background region. When the object being tracked and background has similar statistical property, it is difficult to track the boundary accurately without adding additional constraints. As shown in Fig.(5), applying equation (16) can overcome the problem of misclassification and recover the body part.

5 Conclusion

In this paper, we propose a real time region-based contour tracking algorithm using the fast level set method. The external speed function is constructed based on Chan-Vese Model with the adaptive weighting parameters, λ_i. The main advantage of our algorithm is the ability of tracking objects with weak boundary in real time. The weighting parameters, λ_i are adjusted to accommodate the situations in which the object being tracked and the background have similar intensity. Experimental results demonstrate that our proposed method can overcome the problem of misclassifications by adjusting the weighting parameters.

However, in practical conditions, the low contrast region problem may not be solved by only adjusting the weighting parameter, λ_i. A more sophisticated speed function model will be required to achieve the more robust performance for tracking low contrast objects and it is aimed as our future work.

Acknowledgements

The first author would like to thank Institute for Infocomm Research, Singapore for their financial support.

References

1. Yilmaz A, Xin Li, and Mubarak Shah. Contour-based tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11), November 2004.
2. S. Besson, M. Barlaund, and G. Aubert. Detection and trackign of moving objects using a new level set based method. *Proc. International Conference on Pattern Recognition*, 3:1100–1105, 2000. September.
3. Stauffer C and Grimson W.E.L. Adaptive background mixture models for real-time tracking. *IEEE Comptuer Society Conference on Computer Vision and Pattern Recognition*, 2, 1999.
4. T. Chan and L. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, February 2001.
5. Daniel Cremers. Dynamical statistical shape priors for level set based tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2006.
6. http://www.nlpr.ia.ac.cn/English/irds/gaitdatabase.htm.
7. M. Kass, A. Witkin, and D. Terzopolous. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.
8. A. Mansouri. Region tracking via level set pdes without motion computation. *IEEE Trans. Pattern Analysis and Machine Intelligence,*, 24:947–961, July 2002.
9. N Paragios and R. Deriche. Unifying boundary and region-based information for geodesic active tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, Colorado, USA 1999.
10. N Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence,*, 22(3):266–280, March 2000.
11. Yonggang Shi and W.Clem Karl. Real-time tracking using level sets. *Computer Vision and Pattern Recognition*, June 2005.
12. S.Osher and J.A.Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamiton-jacobi formulations. *Journal of Computational Physics*, 79.:12–49, 1988.
13. S. Zhu and A. Yuille. Region competition: Unifying snake/ballon, region growing, and bayes/mdl/energy for multi-band image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence.*, 18(9):884–900, 1996.

Dynamic Perceptual Quality Control Strategy for Mobile Multimedia Transmission Via Watermarking Index*

Chin-Lun Lai

Communication Engineering Department of Oriental Institute of Technology
58 Sze-Chuan Rd. Sec 2, Pan-Chiao City, 220, Taipei, Taiwan, R.O.C.
fo001@mail.oit.edu.tw

Abstract. In this paper, a simple and efficient strategy of quality control for mobile multimedia transmission is proposed. Based on embedding the watermark into the transmitted media as the channel indication index and using the human visual/auditory sensitivity property, the dynamic bit allocation strategy is adopted accordingly thus to obtain the best quality of the media contents under a fixed bit rate constraint. The simulation results show that by using the proposed channel predication method and bit allocation strategy, not only more transmission efficiency can be achieved but the copyright protection requirement is still met, while only increasing little computational complexity at the mobile terminal.

Keywords: Channel Estimation, Digital Watermarking, Wavelet and Subband Coding, Dynamic Bit Allocation, Human Sensitivity Function, Mobile Multimedia Transmission.

1 Introduction

Channel detection is the key technology of serving the stable and best media quality in wireless transmission applications, especially in cellular system due to the asymmetric calculation power for the base station and mobile terminal and frequency division multiplex (FDD) for uplink and downlink transmission, and is still a challenge for current and future wireless communication system. For example, a GSM base station will transmit a training sequence and receive the analysis feedback signal from terminal as a reference for transmitting power control, and then acquire a better transmission performance consequently. However, more transmission bandwidth is necessary [1]. Rather than using the training sequence, an alternative way [2] uses feedback to inform base station the downlink channel condition, but this approach puts heavily calculation burden on mobile terminal. Moreover, since channel estimation from the training signal in time domain just tells us the global condition rather than each band of the original signal, the optimal strategy to overcome the

* This research was supported by the National Science Council of the Republic of China under NSC 93-2622-E-161-002-CC3.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 712 – 721, 2006.

channel effect corresponding to different frequency band is unavailable. Once the channel condition is known, it is possible to use many new developing schemes, such as transmit diversity [3][4] (open loop) , dynamic modulation scheme, variable bit rate coding, multilevel coding scheme [5][6], … etc., to combat wireless channel effect thus a more efficient communication system is obtained.

On the other hand, due to the fast development of digital watermarking in multimedia transmission, and media sharing service such as 4G mobile communication or P2P service, digital watermark becomes a broad technique for copyright protection, authentication, broadcasting monitoring, in the multimedia transmission applications [7]-[11]. It is noted that since the watermark can be thought as the side information of the transmitting signals [9]-[12], it is reasonable to expect that it is possible to look at the watermarking as useful side information in the communication system to determine the channel condition, ex. the state of the channel noise level. Thus, watermarking not only can be treated as intellectual property protection technique but also performs as a channel indicator in the growing wireless multimedia networks network service. Although this concept had been disclosed previously, there is no explicit way to correlate the received watermark and the channel condition. It is expected that via calibrating the distortion rate of the transmitted watermark and original signal, the channel condition can be estimated precisely by a simple method. After the channel condition is known, the specific coding scheme for different channel condition can be adopted to improve the transmission efficiency for both the bit error rate (BER) and transmission rate.

This paper presents an effective approach, which is expected to be satisfied in both respects: copyright protection and channel effects compensation, for mobile multimedia transmission. By detecting the watermark information embedded in every subband of the media, the channel condition can be easily estimated and feedback to the transmitter, then a suitable coding algorithm is adopted to overcome the channel effect. The human visual/auditory sensitivity property is used in the proposed bits allocation scheme between different bands and source/channel coding aspect [13], thus to achieve the goal of quality control under limited bit rate criterion, which is important in mobile transmission such as 3G/4G multimedia service. For example, for those important bands, more bits should be assigned to overcome the severe channel effect to obtain the acceptable image quality. However, if the band condition is indicated as "good enough", the extra bits can be reallocated to other bands with more distortion thus a better image result with fine detail is available.

The paper is organized as follows. In section 2, the theoretical principle and coding algorithm of the proposed strategy is described. The simulation results and discussions are given in the section 3, and finally, the conclusion and the future work are given in the section 4.

2 Theoretical Principles and Algorithms Description

The simplified block diagram of the proposed scheme is shown in Fig. 1 and is described as follows.

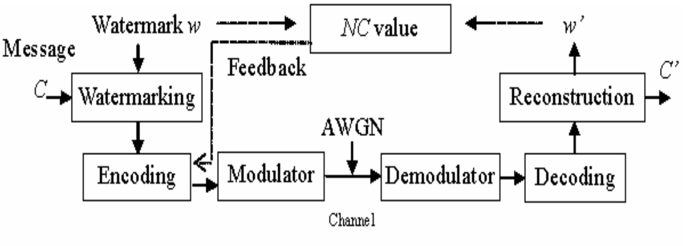


Fig. 1. Simplified block diagram of the proposed transmission system. Note that *C* represents the original subband coefficients.

2.1 Watermark Embedding and Detection

It is observed that since the time domain training sequence is used in the general transmitting systems, it is hard to indicate channel condition accordingly to subband individually. To gain the better efficiency from coding strategy, a subband decomposition codec with watermarking in each band is proposed as follows. Let C_o denotes the original source to be transmitted, then a wavelet decomposition filter pair with impulse response (h_{dL}, h_{dH}) is used to divide the baseband signal into two subbands, says C_L and C_H band respectively, by

$$C_L = C_o * h_{dL}, \quad h_{dL} = [0.707, 0.707] \tag{1}$$

$$C_H = C_o * h_{dH}, \quad h_{dH} = [0.707, -0.707] \tag{2}$$

where h_{dL} and h_{dH}, also known as Daubechies wavelets with N=1, and (C_L , C_H) represent the high and low frequency bands respectively. The reconstruction process is the same as in (1)-(2) by convolving the decomposed bands with the reconstruction filters h_{rL} and h_{rH}, which are represented as $h_{rL}= h_{dL}$ and $h_{rH}= - h_{dH}$. For an image source C_o, the wavelet analysis is applied in the same manner two times for each row and column thus to obtain four subband coefficient images C_{LL}, C_{LH}, C_{HL}, and C_{HH}, respectively.

The watermarking technique we used is similar to Hsu's method [14] by using the DWT and multiresolution technique, which is stated in brief as follows. For each wavelet band, a pre-determined binary watermark template with ¼ size of the coefficients image is undergone the successive layers decomposition, pixel-based pseudo permutation, and block-based image dependent permutation processes. After that, the watermark information is embedded into the transform domain by modifying the DWT coefficients according to their neighboring relationship. For example, the successive two middle-frequency coefficients are compared by subtraction, if the former is greater, it denotes that a bit '1' can be embedded directly. On the contrary, if a bit '0' is to be inserted, a number must be added into the later coefficient thus to make it greater than the former. An example result for watermark hiding is shown in Fig. 2 by using the above watermarking algorithm

The watermark can be self extracted without the original image [14]. The received image is undergone the same processes of wavelet decomposition, then the recovered

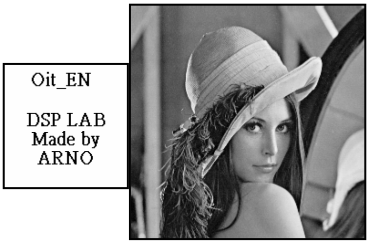


Fig. 2. (Left) Binary text watermark image of size (128*128), (Right) The resultant image after watermarking (256*256)

watermark can be obtained by comparing the neighboring coefficients in middle-frequency to disclose the transmitted bit symbols, and descrambling the bit stream to reconstruct the watermark image.

2.2 Channel Estimation by Watermark Correlation

Due to the stochastic property of the original message, it is reasonable to estimate the channel condition by comparing the original and the reconstructed watermark. Thus, after the watermark information is extracted from the terminal, a simple and effective operation can be performed to estimate the channel condition for each frequency band. To reduce the computational complexity, a normalized correlation (*NC*) coefficient is used to measure the fidelity of the reconstructed watermark as follows

$$\mathrm{NC} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} w(i, j) w'(i, j)}{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [w(i, j)]^2} \tag{3}$$

where *w* and *w'* denotes the original and received watermark respectively. The *NC* coefficient reveals the information of the channel effect for a communication system, for example, little noise and channel distortion made the *NC* near to one, while heavy noise or severe channel distortion made it approach to zero. Thus, a simple indicator of the channel condition can be obtained by observing the *NC* coefficient and referring to a pre-determined lookup table, which correlates the *NC* value to the noise level (signal to noise ratio: *SNR*, or noise variance: σ^2) or bit error rate (BER) of the communication system. A typical *SNR* definition is given as follows

$$\mathrm{SNR}=10\ \mathrm{Log} \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [C_o(i, j)]^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [C_o(i, j) - C'(i, j)]^2} \tag{4}$$

where *C'* denotes the noisy message. When a noise with known power variance is added to the transmitted image, a corresponding *NC* value from the corrupted watermark can be calculated by (3), and the corresponding *SNR* value can be determined by (4). A simulation result for two popular noise models, Gaussian and Uniform, is given in Fig. 3 to describe the relevance between the *NC* and *SNR*.

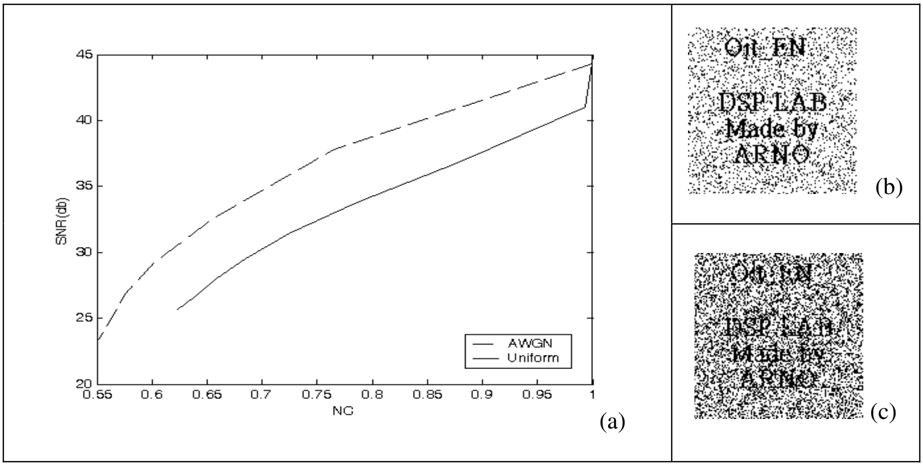


Fig. 3. (a) Correlation curves of *NC* and *SNR* values for Gaussian and Uniform noise models. (b) and (c) show the recovered watermark corresponding to receiving an 'acceptable' or 'bad' transmitted media (Lena) quality respectively. This figure illustrates that the received media fidelity can also be predicted by observing the received watermark.

According to this result, the *NC* and *SNR* can be correlated approximately (note that numerical errors will appear at *NC* near 0 and 1 since using curve fitting) by

$$SNR\_{\text{Gaussian}} = -142.8NC^2+254.5NC-71.9 \quad \text{(dB)} \tag{5}$$

Thus, it is easy for us to know the channel noise power level just by compute the *NC* at terminal side.

2.3 Optimal Bit Allocation Strategy by Human Perception Sensitivity Property, Image Contents, and Channel Condition

Once the channel condition for each band is known, it can be sent back to the transmitter as the control signal to adjust the system parameters, thus the extra transmission efficiency can be achieved. In this paper, a dynamic bit allocation scheme for each signal band under a fixed bit rate criterion is used to improve the visual/auditory quality of the transmitting media under various channel conditions. That is, to make sure that the satisfactory quality for various channel condition can be achieved.

To achieve this goal, the threshold points, which indicate the maximum number of removable bits, corresponding to 3 kinds of visual/auditory quality level must be determined by a series of subjective experiments. A number of members in our research group are selected to join this test. The wavelet coefficients of the transmitted media are removed (or replaced by zero) one by one according to the visual/auditory weighting function, which is similar to the concept of HVS-quantization method in wavelet [15]. That is, wavelet coefficient with lower

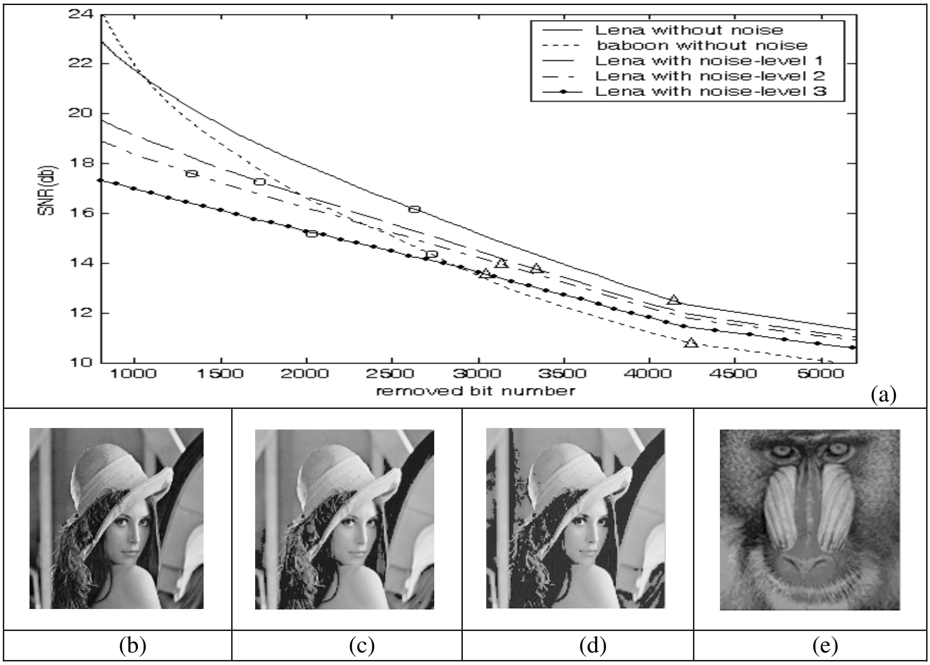


Fig. 4. (a) Threshold points of 'good--□', and 'acceptable--△' for Lena image with different noise levels. It indicates the maximum removable bits to obtain the desired image quality and the resultant *SNR* value. (b)-(d) represent the results of 'good', 'acceptable', and 'bad' quality in noiseless condition. The noise level 1, 2, and 3 are 24.9, 99.2, and 404.7 (dB) respectively. Note that a high frequency content image 'Baboon' is also tested with the same condition to (b) and is shown in (e) for comparison, and the completed test results are depicted as a dotted curve in (a) for the Baboon image under noiseless condition.

perceptual weighting should be removed first. Two threshold points: 'good' and 'acceptable' are decided during the degradation process if most of the testers observe the significant quality change between the successive images. That is, in the previous image we feel "good enough" and in the next image we feel "not good". Resultant curves of two test examples, 'Lena' and 'Baboon' image, at different noise level are shown in Fig. 4(a) to indicate the removed bit number and the received image fidelity. The threshold points are also examined and the corresponding images for noiseless channel case are shown in Fig. 4(b)-(d), which represents the 3 categories images of the visual quality requirement. In our coding strategy, the concept of 'bit allocation' includes two aspects and depends on the channel condition and the required service quality. If the communication system is undergone a more noisy or distorted channel, more bits should be allocated to channel coding algorithm to protect perceptually important coefficients. However, if the channel effect is negligible, more bits can be used in the source coding algorithm to improve the fine detail of the transmitted media.

For example, if C_{LL} band is corrupted severely, it means the acceptable quality is unavailable; we must remove some bits, from other bands, to protect C_{LL} by channel coding techniques until the desired quality result is reached. On the other hand, if the channel condition is 'good enough', more bits, which exceed the quality requirement for a specific desired service, can be allocated to C_{LH}, C_{HL}, and C_{HH} bands to get the higher image quality with fine detail. Thus, the number of maximum removable bits, which depends on both the image contents and the channel condition, gives us the guiding principle for tuning the coding parameters.

To simplify the coding process, the image content classification is represented by the power ratio of the high frequency to the full band, says C_{HH}/C. According to the previous experimental results, the number of maximum removable bit number corresponding to different channel conditions and the desired quality can be approximated by

$$R\_{good} = 665+3700x+6727x^2-5.451y+0.0093602y^2 \tag{6}$$

$$R\_{accept} = 1690+3298.5x+4880x^2-9.5704y+0.017058y^2 \tag{7}$$

where x denotes the power ratio of the high frequency band to the overall bands, and y denotes the noise level in variance respectively. The above two equations, simply give us the information of minimum required bits number at the transmitter when the noise level and the desired service quality are known.

Audio signal are also tested in the same process. For audio signal, the binary watermark image is embedded into frequency band coefficients with less auditory sensitivity. The optimal threshold points for removable bit number, similar to (6)-(7), are represented by

$$R\_{good} = 4158+787x+149x^2-21.05y+0.048951y^2 \tag{8}$$

$$R\_{accept} = 9401+1779.6x+337x^2-34.391y+0.07295y^2 \tag{9}$$

3 Simulation Results and Discussions

To evaluate the performance of the proposed system, a designed MATLAB procedure with Simulink block is used to simulate the whole process of the communication system, the watermark signal is the same as in Fig. 2 and the test images are Lena, Baboon, …etc. After the wavelet decomposition and watermarking process, each watermarked frequency band C_{wi}, i=1~4, is undergone the digital modulation scheme and is transmitted over a wireless channel with a specific noise models added in. At the receiver terminal, under the perfect sample-timing assumption, the output of the matched filter is C_{wi}', i=1~4, and then the corresponding reconstruction algorithm is used to retrieve the watermark from the received signal.

The channel condition in each band can be known by directly calculating the *NC* coefficients by (3) and then referring to (5) to obtain the noise power. After that, the channel condition index and the power ratio of the high/overall frequency band, are used as the feedback signal to the transmitter. Once the desired service quality is assigned, by referring to (6)-(9), it is possible to allocate bits precisely either between different signal bands or channel/source coding schemes, to obtain the best perceptual quality media under the present channel condition. To simplify the coding process, the coding scheme is emulated by preserving the corresponding number of the original image coefficients according to the bit allocation result. One example result of the complete simulation process at different channel condition is shown in Fig. 4 and Fig. 5. It can be observed in Fig. 4(a) that for the same image under different channel conditions, the removable bits increases if the channel noise level decreases. On the other hand, it also reveals that for two different images under the same channel condition, the removable bits number is greater for the image with higher high-frequency power. These results tell us that more bits can be (or should be) saved and re-allocated to the channel coding algorithm to protect those perceptually important coefficients during transmission. In Fig. 5, the resultant images and the corresponding *SNR* values by using or not using the dynamic bit allocation strategy (total allocated bits=52Kb) for three noise levels, are shown in Fig. 5(a)-(f) and the figure caption. From these results, it is observed that at a fixed bit rate transmission constraint, which is often used in general wireless transmission service, the proposed algorithm not only save bits by dropping the training sequence, but also achieve the comparable image quality than the innovative system [16]. Something interesting should be noted in Fig. 5(d) and (f) where the image with lower *SNR* in (f) seems has the better visual quality than (d), however, it is trivial that sometimes severe channel noise condition overcomes the contouring effect, which comes from the heavy quantization, and then results a visually smooth image. Nevertheless, this contradiction disappears if the dynamic bit allocation scheme is used (shown in (a) and (c)).

Moreover, it should be noted that the reconstructed watermark image can be still recognized if the received source media quality is not below the acceptable point, thus is able to maintain the copyright protection requirement. On the other hand, by using the hierarchical coding structure and the human sensitivity property, the channel effects such as delay or lost packet, which often damage the mobile transmission system, can be alleviated significantly. For example, if the channel is known to be not

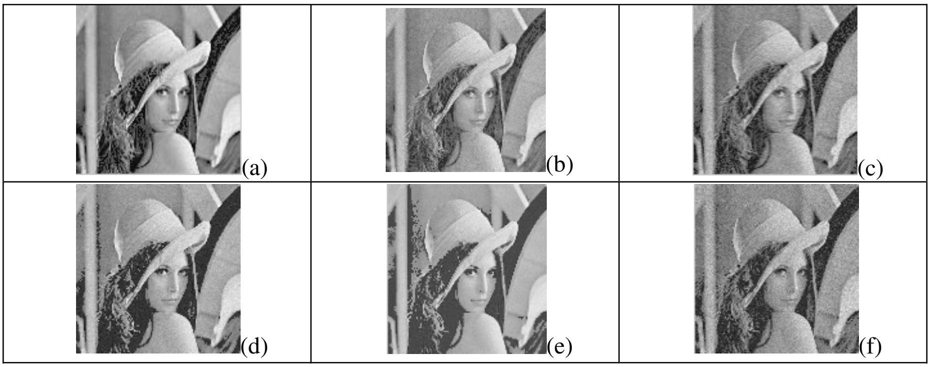


Fig. 5. Resultant images with (abc) and without (def) using the dynamic bit allocation scheme under different channel noise levels. (a)(d): ^{2}=24.9, (b)(e): ^{2}=99.2, and (c)(f): ^{2}=404.7. Each image is allocated with 52K bits. The *SNR* for (a)-(f) are 15.1, 14.8, 14.3, 13.1, 12.9, and 12.2 (dB) respectively.

so good, the “most important part” of the media contents will be protected by more channel coding symbols thus to overcome the lost packet effect and maintain a minimum satisfactory media quality.

Other media type such as video is also tested, although the video case hasn’t been simulated completely, most test examples also give us the similar conclusions.

Finally, different noise models, like Rayleigh or Ricean, give the similar simulation results. Although the channel model is more complicated in a real transmission system, it is feasible to use the similar algorithm to detect and compensate the unknown channel effect thus to meet the quality control requirement.

4 Conclusions and Future Work

An intuitive scheme for channel detection and optimal bit allocation, for best perceptual media quality in mobile multimedia transmission under fixed bit-rate constraint, is proposed. Via the watermarking technique as the channel indication, the training sequence, which is used in the general mobile communication system to estimate the channel condition, can be omitted thus more resource can be saved. The proposed system not only fits the demand of copyright protection of digital contents but also increase the transmission efficiency and the perceptual quality within a bandwidth limitation. Moreover, detecting the channel condition by watermarking in frequency domain offers the ability to examine the channel for each individual subband, thus is able to overcome the different channel effects for the corresponding band. The simulation results show the good performance of the proposed scheme in receiving visually better images than before, especially in an extreme condition with severe noisy environment. That is, the proposed system can provide a more perceptually satisfactory media quality than the previous systems which may have the unacceptable results, thus is much practical for 3G/4G multimedia service like H.324M mobile video telephony applications.

Finally, although it is reasonable to expect that the implemented system will exhibit a good work by observing the simulation result, modifications may be necessary in real system to obtain the optimum performance according to various conditions.

Various skills are helpful to enhance the performance of the transmission system by using the similar concept and are left as the future works. For example, the multilayer wavelet decomposition method with various wavelet functions, may further improve the efficiency of the bit allocation scheme. Moreover, various watermarking or coding schemes including channel coding and source coding, can be test to find the optimal system framework for different conditions. Of course, skills including digital modulation or transmit diversity are also encouraged to be examined to enhance the system performance further.

Acknowledgments. C.L. Lai would thank to the group members of Dr. J. Huang and J.C. Yang for their help on part of the MATLAB simulation and comment.

References

1. Jorg Eberspacher, Hans-Jorg Vogel: GSM: Switching, Service and Protocols. John Wiley & Sons (1998)
2. A.F. Naguib, A. Paulraj, T. Kailath: Capacity Improvement with Base Station Antenna Arrays in Cellular CDMA. IEEE Transactions on Vehicular Technology, vol. 43, no. 3 (1994) 691-698
3. S.M. Alamouti: A Simple Transmit Diversity Technique for Wireless Communications. IEEE Journal on Select Areas in Communication, vol. 16 (1998) 1451-1458
4. Josef Fuhl, Alexander Kuchar, Ernst Bonek: Capacity Increase in Cellular PCS by Smart Antennas. IEEE VTC Conference, vol. 3 (1997) 1962-1966
5. M. El-Sayed, J. Haffe: A View of Telecommunications Network Evolution. IEEE Communications Magazine (2002) 74-81
6. John G. Proakis: Digital Communications. 4th Edition, McGraw-Hill International Edition (2001)
7. J. Sun, J. Sauvola, D. Howie: Features in Future: 4G Vision from a Technical Perspective. IEEE2001
8. J. Borras-Chia: Video Service over 4G Wireless Networks: Not Necessarily Streaming. IEEE2002
9. T. Kalker, at el.: Music2Share - Copyright-Compliant Music Sharing in P2P Systems. Proceedings of the IEEE, vol. 92, no. 6 (2004) 961–970
10. I. Cox, J. Bloom, M. Miller: Digital Watermarking: Principles & Practice. Morgan Kaufmann Publishers, Inc. San Francisco (2002)
11. G.C. Langelaar, I. Setyawan, R.L. Lagendijk: Watermarking Digital Image and Video Data - A State-of-the-Art Overview. IEEE Signal Processing Magazine, vol. 17, no. 5 (2000) 20-46
12. C.E. Shannon: Channels with Side Information at the Transmitter. IBM J. Res. Develop. vol. 2 (1958) 289-293
13. C.L. Lai: The Dynamic Detection and Optimal Codec Allocation for Wireless Transmission based on Digital Watermark Calibration. Grant Project Report of National Science Council of the Republic of China, NSC 93-2622-E-161-002.
14. C.T. Hsu, J.L. Wu: Multiresolution Watermarking for Digital Images. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no. 8 (1998) 1097-1101
15. M.J. Nadenau, Julien Reichel, Murat Kunt: Wavelet-Based Color Image Compression: Exploiting the Contrast Sensitivity Function. IEEE Transactions on Image Processing, vol. 12, no. 1 (2003) 58-70
16. D.M. Chandler, S.S. Hemami: Multiresolution Watermarking for Digital Images. IEEE Transactions on Image Processing, vol. 14, no. 4 (2005) 397-410

A Rate Control Using Adaptive Model-Based Quantization

Seonki Kim[1], Tae-Jung Kim[2], Sang-Bong Lee[2], and Jae-Won Suh[2]

[1] LG Electronics, Inc.,
Seoul, Korea
seonki@lge.com

[2] Chungbuk National University, School of Electrical and Computer Engineering,
12 Gaeshin-dong, Heungduk-gu, Chongju, Korea
taejung@cbnu.ac.kr, sblee@cbnu.ac.kr, sjwon@cbnu.ac.kr

Abstract. The rate control is an essential component in video coding to provide a uniform quality under given coding constraints. The source distribution to be quantized cannot be defined as a single model in the video sequence. In this paper, we present a new rate control algorithm based on the Generalized Gaussian R-D model. Through considering a relation between adjacent frames or macroblocks, we determine model parameters, and perform a rate control on the H.264/AVC video codec. As shown in experimental results, the proposed algorithm provides an improved quality of the reconstructed picture after encoding. In addition, our scheme generates the number of bits close to the target bitrate.

1 Introduction

Generally, the output bitrate for video coding is controlled by the quantization parameter that is determined in the encoder. In other words, the selection of the proper quantization parameter controls the number of output bitrate and provides the best quality within coding constraints, such as target bitrate and frame rate, etc. In addition, the rate control is also required to ensure a prevention of overflow or underflow of the buffer. The loss of data by overflow that is an irreparable loss leads to the degradation of the quality after encoding. Therefore, the rate control is an essential component in video coding. However, the rate control itself is not specified by the standard, and is only considered in the encoder side. Although rate control algorithms are not mandatory for video coding standard, video coding standards have own rate control models, such as MPEG-2 TM5 [1], MPEG-4 Q2 model [2], and H.263 TMN8 [3].

The classical rate-distortion model based quantizer has been studied for a long time. However, at low bitrate, it doesn't work well since the variance of the source data was only considered as a parameter to control rate. The variance itself is not sufficient to characterize source data at low bitrate. In order to overcome this problem, rate-quantization models that generate the bitrate close to the target bits have been studied in [6,7,8]. These rate control algorithms using rate model that is derived from the rate-distortion function or rate-quantization

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 722–731, 2006.

curve assume that the data to be quantized was considered and represented as having a single distribution. In the realistic case, however, all frames cannot be unified with a unique model.

In this paper, we propose a new rate control algorithm considering the adaptive model adjustment. Based on the characteristics of the Generalized Gaussian distribution, as source statistics, rate control is done adaptively. Using previous coding information and the current coding mode, we decide model parameters and estimate a source statistic. From the experimental results, we conclude that the proposed algorithm provides an acceptable and improved quality, and guarantees a generation of bitrate close to the target bitrate.

2 Proposed Algorithm

2.1 Rate-Quantization Model

We consider R-D function of the Generalized Gaussian function:

$$R(D) = \frac{1}{\gamma}\log_2\left(\frac{\sigma^\beta}{D}\right) \tag{1}$$

where σ denotes the standard deviation of the source data to be quantized, and β and γ are model parameters. Previous works [9,10] show that the Generalized Gaussian distribution is suitable for designing rate model about DCT coefficients. β is a free parameter controlling the exponential rate of skewness of distribution. As special cases, when $\beta = 1, \gamma = 2$ and $\beta = 2, \gamma = 1.5$, the Generalized Gaussian distribution can be regarded as the Laplacian and Gaussian distributions, respectively.

Before designing a rate model, we should consider the amount of the distortion as the change of quantization parameter (QP) because information loss is done by QP. In general, the distortion by the quantization is defined as follows,

$$D(Q) = c \cdot Q^2 \tag{2}$$

where c is the distortion parameter that represents a relationship between distortion and quantization parameter. The value of distortion parameter gives an inspiration to make a region-based rate control scheme. We do not handle it in this paper. The decision of distortion parameter will be described in the next section. From Eq. (1) and Eq. (2), rate-quantization model is formulated as:

$$Q = \sqrt{\frac{\sigma^\beta \cdot 2^{-\gamma R}}{c}} \tag{3}$$

In this rate-quantization model, the selection of parameters, which are β, γ, and c, is important in the rate-quantization model, as their values presents a different model that reflects the characteristics of source data. More concrete description of the estimation of model parameters will be given in the next section.

2.2 Estimation of Coding Parameters

In section 2.2, we designed a new rate-quantization model based on the rate-distortion model of Generalized Gaussian distribution. The Eq. (3) requires 5 parameters: shape factor (β), distortion parameter (c), standard deviation of source (σ), target rate (R), and model parameter (γ). The source statistic value that is the standard deviation is explained in the above section how to get it from the structural constraint of H.264 video coding. In this section, we will explain that the methods to obtain other coding parameters required in the proposed algorithm.

Shape Factor. The shape factor β is an important parameter to model a form of the source distribution. However, the exact estimation of the shape parameter for the source is not easy. It also requires a high computational complexity. In order to avoid this problem, some researches have been studied to find the optimal shape factor with small complexity [9,10].

In this paper, we simply decide the shape factor by a similarity between two adjacent images.

$$\beta = \begin{cases} 1, & \omega > 0.9 \\ 2, & \omega < 0.6 \\ 2 - 3.3 \cdot (\omega - 0.6), & a \leq \omega \leq b \end{cases} \tag{4}$$

$$\omega = \frac{1}{256}\sum_{i=1}^{256} v(i), v(i) = \begin{cases} 1, & |x - \widehat{x}| < 2 \\ 0, & \text{o.w.} \end{cases} \tag{5}$$

where x and $\widehat{x}$ denotes the samples, which are placed on the corresponding point, in the current and the reference frames, respectively. In this paper, we assume that the source distribution is modeled as one of forms existing between Laplacian, that is defined by the value of 1, and Gaussian, that is defined by the value of 2.

Distortion Parameter. In Eq. (2), we design a distortion model by a distortion parameter, c, and a quantization parameter, Q. As a quantization parameter, the distortion is also considered at the macroblock-level. In order to decide the distortion parameter for each macroblock, we consider the typical distortion measure of quantization [3].

$$D = \frac{\alpha}{3}Q^2 \tag{6}$$

where α is a distortion weight for the current macroblock. From Eq. (2) and Eq. (6), we can derive the equation to compute a distortion parameter, c, as follows:

$$c = \frac{\alpha}{3} \tag{7}$$

$$\alpha = \begin{cases} \frac{B_T}{\sigma \times (256N)}, & \frac{B_T}{256 \times N} < \frac{1}{2} \\ 1, & o.w. \end{cases} \tag{8}$$

where B_T denotes the target bits for a frame, σ is the standard deviation. N is the number of macroblocks in a frame.

Model Parameter. We should update the model parameter γ. Let R_T be the target bits and R_A be the coding bits per pixel. Thus, R_T and R_A is obtained by a division by number of samples in the frame. Theoretically, R_T follows the rate-quantization model because the rate-quantization model was based on the rate-distortion function. Thus, we can express R_T and R_A as follows:

$$R_T = \gamma_e \log_2 \frac{\sigma^\beta}{c \cdot Q^2}, \qquad R_A = \gamma_c \log_2 \frac{\sigma^\beta}{c \cdot Q^2} \tag{9}$$

where γ_e and γ_c are the model parameter used for the current frame and the model parameter to be used for the coming frame.

$$R_T - R_A = \left(\frac{1}{\gamma_e} - \frac{1}{\gamma_c}\right) \cdot \log_2 \left(\frac{\sigma^\beta}{c \cdot Q^2}\right) = \left(\frac{1}{\gamma_e} - \frac{1}{\gamma_c}\right) \cdot A \tag{10}$$

$$\therefore \gamma_c = \frac{A\gamma_e}{A - \gamma_e \left(R_T - R_A\right)} \tag{11}$$

where

$$A = \log_2 \left(\frac{\sigma^\beta}{c \cdot Q^2}\right) \tag{12}$$

The value of A is small.

The model parameter is obtained at the frame-level, but other parameters are obtained at each macroblock-level. In order to define the source statistic for the frame, we re-calculate for the actual residual data that is obtained after coding. Because the model parameter is updated after the coding for the frame, we can use a real residual data. The distortion parameter and the shape parameter for frame-level is obtained as follows:

$$c\left(\beta\right) = \frac{1}{N} \sum_{i=1}^{N} c_i \left(\beta_i\right) \tag{13}$$

where N is the number of macroblocks in a frame, and c_i and β_i indicate the average distortion parameter and the average shape factor for the frame, respectively.

2.3 Computation of Quantization Parameter

We compute the quantization parameter for the current macroblock by the designed rate-quantization model shown in Eq. (3). The calculated quantization parameter is rounded to nearest integer value. The quantization parameter is also adjusted as follows in order to reduce a remarkable quality variation between the current macroblock and the previous macroblock.

$$QP^* = \begin{cases} QP_{prev} - 2, & \text{if } QP - QP_{prev} < -2 \\ QP_{prev} + 2, & \text{if } QP - QP_{prev} > 2 \\ QP, & \text{otherwise} \end{cases} \tag{14}$$

The QP^* is the adjusted quantization parameter. After an adjustment of quantization parameter, we should set to the quantization parameter value in the range of 0 to 51. This range is defined in the standard of H.264 video coding.

2.4 Rate-Distortion Optimization and Its Constraint in H.264

Causality Problem. H.264 introduced the rate-distortion optimization technique to increase the coding efficiency. The rate-distortion optimization is based on Lagrangian optimization method. H.264 defined the formula to calculate the Lagrangian multiplier.

$$\lambda = 0.85 \times 2^{QP/3} \tag{15}$$

In λ computation, QP value is required, and λ is calculated in a initial coding process in H.264 video coding order. As shown in Eq. (15), the computation of the Lagrangian multiplier requires the value of the quantization parameter. However, the decision of the quantization parameter needs the standard deviation of source data. Therefore, we get into difficulty for the "causality problem" or "chicken-and-egg dilemma." In order to avoid that limitation in the H.264 video coding, we should predict the standard deviation value.

Prediction of Standard Deviation. For the causality problem, Ning [11] proposed one method:

$$\sigma = \frac{8 \times \sigma_P + \sigma_P^L + \sigma_P^T}{10} \tag{16}$$

where σ is the estimated standard deviation for the current macroblock and σ_P is an actual standard deviation of the macroblock in the previous frame. σ_P^L and σ_P^T are the standard deviations of the left and top macroblock of the macroblocks placed at the same position in the previous frame. Ning's method has a problem which depends on information for the previous frame, entirely. Therefore, Ning's method can include a possibility for the high prediction error because it doesn't consider the characteristics of the current frame itself.

In order to reflect the spatial relationship within a frame when we predict the standard deviation, we consider the standard deviation of the adjacent macroblocks of the target macroblock. The proposed predictive method is as follows:

$$\sigma = \begin{cases} \alpha_1 \times \sigma_P, & \dfrac{\sigma_C^L + \sigma_C^T}{2} > \alpha_1 \times \sigma_P \\ \alpha_2 \times \sigma_P, & \dfrac{\sigma_C^L + \sigma_C^T}{2} < \alpha_2 \times \sigma_P \\ \dfrac{6 \times \sigma_P + 2 \times \{\sigma_C^L + \sigma_C^T\}}{10}, & \text{otherwise} \end{cases} \tag{17}$$

where σ_P and σ_C are the standard deviation of macroblock in the previous frame corresponding to the same position for the current frame and the standard deviation for macroblocks in the current frame, respectively. T and L represent the top macroblock and left macroblock of the current macroblock, respectively. Two parameters, α_1 and α_2, are set by 1.1 and 0.9, respectively. These values are heuristically decided by experiments.

3 Experimental Results

3.1 Simulation Condition

We implement the proposed algorithm in the H.264 reference software. In order to evaluate the proposed quantization algorithm, we set a common simulation conditions that follow the baseline profile. The baseline profile is specified in the H.264 standard document [13]. Table 1 shows our simulation conditions.

Table 1. Simulation Conditions

Rate-Distortion Optimization	On
GOP Structure	IPPP
Symbol Mode	CAVLC
MV search range	32
Reference Frames	1

The proposed algorithm is applied for the P-frames because B-frame coding is not included in the baseline profile. For the I-frames, we use the fixed quantization parameter. Test video sequences used for simulation are "Foreman", "News", "Silent" and "Mother and Daughter" sequences. These sequences are defined as the test sequences in the document. The resolution of test sequences is QCIF with 4:2:0. Target bits for experiments are 32, 48 and 64 kbps and frame rates are 10 fps.

First of all, we verify the accuracy of the proposed prediction method of standard deviation for source. Figure 1 shows the comparison between the original standard deviation and the predictive standard deviation by the proposed method. The predictive values are obtained by Eq. (17). The proposed method generates the standard deviation close to the original value.

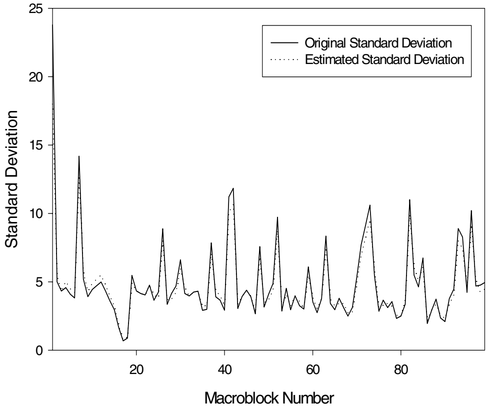


Fig. 1. Standard deviation between the original value and the predictive value

3.2 Performance Evaluation

In order to evaluate the proposed quantization algorithm, we compare test results with the MPEG-2 TM5 [1], H.263 TMN8 [3], and Siwei's algorithms [4].

Table 2 shows the comparison between the number of coding bits for the proposed algorithm and target bits. As shown in Table 2, the proposed algorithm generates the number of bits close to the target bits within 1% difference.

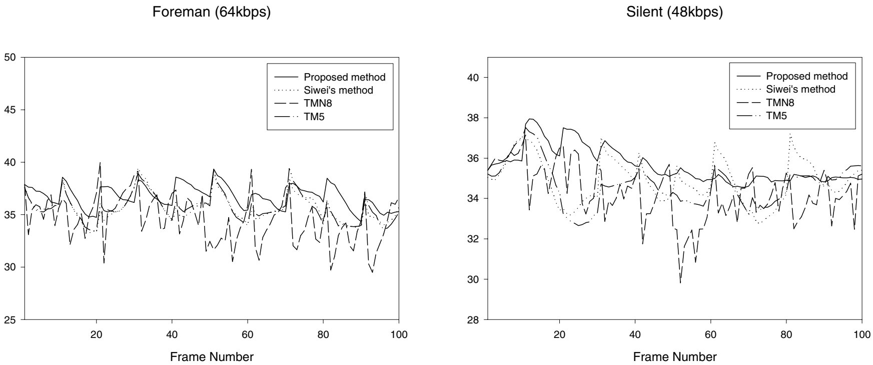
Table 2. Coding bits and target bits for the proposed rate control (kbps)

Test Sequence	Target bits	Coding bits	Difference
News	32	31.83	-0.17
Foreman	32	32.08	0.08
Silent	48	48.06	0.06
MAD	48	48.34	0.34

One of measures to show the performance is PSNR. Table 3 describes the comparison of the average PSNR values for the proposed algorithm and others. The proposed scheme provides the considerably good results. The uniform PSNR service is also required to reduce the quality variation in terms of human visual scene. Figure 2 describes PSNR fluctuations for two sequences. As shown in

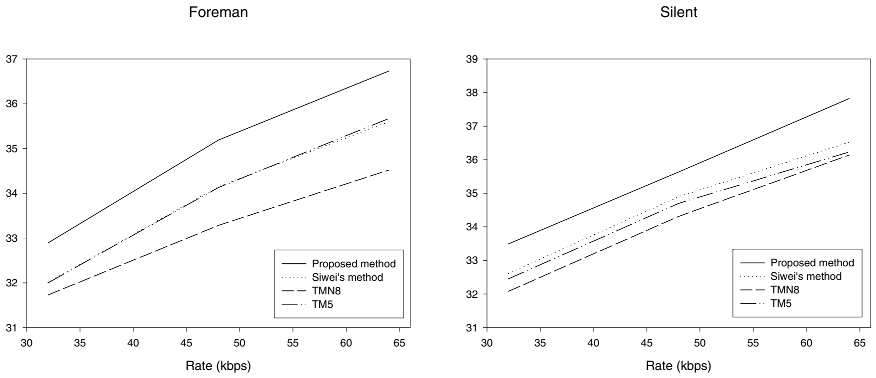
Table 3. Comparison of average PSNR among three rate control algorithms (dB)

Test Sequence	Target bits (kbps)	Proposed algorithm	Siwei's algorithm	TMN8 algorithm	TM5 algorithm
News	32	34.49	33.48	33.00	33.45
Foreman	32	32.89	32.01	31.73	32.00
Silent	48	35.64	34.90	34.31	34.70
MAD	48	40.04	39.57	38.50	39.19



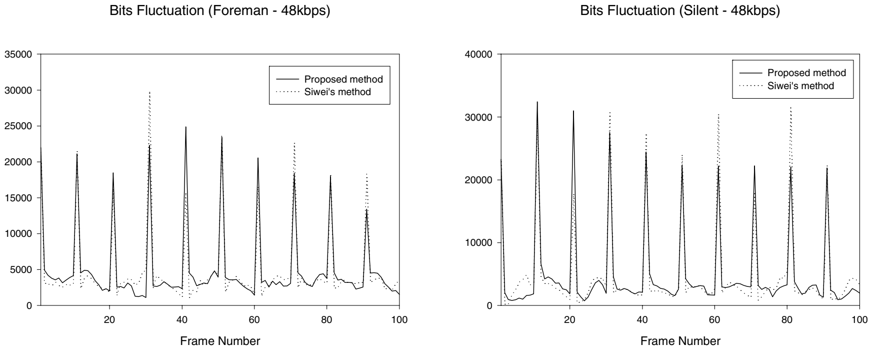
(a) Foreman, 64kbps (b) Silent, 48kbps

Fig. 2. PSNR fluctuation comparison



(a) Foreman (b) Silent

Fig. 3. Rate-distortion curve, Target rates (32, 48, and 64 kbps)



(a) Foreman, 48kbps (b) Silent, 48kbps

Fig. 4. Bits fluctuation comparison between the proposed and Siwei's algorithms

Figure 2, the proposed algorithm provides the improved reconstructed image quality without large quality variation rather than others.

In Figures, we show results for two kinds of test sequences that are "Foreman," and "Silent." The reason for using two sequences is that two sequences have the conflicted characteristics. "Foreman" has high complexity in terms of motion and texture composition rather than "Silent" sequence.

Figure 3 shows the rate-distortion curve for three kinds of target bits that are 32, 48, and 64kbps. As shown in Figure 3 clearly, the proposed algorithm provides higher coding efficiency than others.

One of main applications in H.264 is the mobile communications. The general mobile communication has a low bit-rate channel or has a small capacity to transmit a video data. At the low bitrate, small bit fluctuations are required to avoid an abrupt channel use. Figure 4 shows the bit fluctuations for two sequences at the specific target rates. The proposed algorithm generates the coding bits without an excessive variation.

4 Conclusion

The rate control for H.264 requires a low computational complexity since H.264 has a high coding complexity. In this paper, we proposed an adaptive model-based rate control scheme for the effective coding of inter-frames. The proposed rate-quantization model is designed from the rate-distortion function of Generalized Gaussian Distribution, which can adaptively represent various distributions by changing of the shape factor. The shape factor is simply obtained from the comparison of difference between macroblocks. The quantization parameter is calculated by the designed rate-quantization model. Simulation results show that the proposed rate control scheme generates coding bits close to the target bits and provides improved coding efficiency at low bit rates.

References

1. MPEG-2, *MPEG-2 Test Model 5(TM5) Doc.ISO/IEC/ JTC1/SC29/WG11/ N0400*, Test Model Editing Committee, Apr. 1994.
2. MPEG-4, *MPEG-4 Video Verification Model version 16 (VM16) Doc.ISO/IEC JTC1/SC29/WG11/N3312*, Mar. 2000.
3. ITU-T SG16 Video Coding Experts Group, *Video Codec Test Model, Near-Term Version 8 (TMN8)*, Sep. 1997.
4. Siwei Ma, Wen Gao, Peng Gao, and Yan Lu: Rate Control for Advance Video Coding (AVC) Standard, *International Symposium on Circuits and Systems*, Vol.2, pp.892-895, May 2003.
5. Ning Wang and Yun He, "A New Bit Rate Control Strategy for H.264," *Joint Conference of the 4th International Conference on Information, Communications and Signal Processing and 4th Pacific-Rim Conference on Multimedia*, 3A2.6, Dec. 2003.
6. Liang-Jin Lin and Antonio Ortega, "Bit-Rate Control Using Piecewise Approximated Rate-Distortion Characteristics," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8, No. 4, Aug. 1998.
7. Jordi Ribas-Corbera and Shawmin Lei, "Rate Control in DCT Video Coding for Low-Delay Communications," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 1, Feb. 1999.
8. Zhihai He and Sanjit K. Mitra, "A Linear Source Model and a Unified Rate Conrtrol Algorithm for DCT Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 12, No. 11, Nov. 2002.
9. Edmund Y. Lam and Joseph W. Goodman, "A Mathematical Analysis of the DCT Coefficient Distribution for Images," *IEEE Trans. on Image Processing*, Vol. 9, No. 10, pp.1661-1666, Oct. 2002.
10. Gregory S. Yovanof and Sam Liu, "Statistical Analysis of the DCT Coefficients and Their Quantization Error," *Conference on the Thirtieth Asilomar*, Vol. 1, pp.601-605, Nov. 1996.
11. N. Wang and Y. He, "A new bit rate control strategy for H.264," Pacific-Rim Conference on Multimedia, Dec. 2003.

12. B. Aiazzi, L. Alparone, and S. Baronti, "Estimation Based on Entropy Matching for Generalized Gaussian PDF Modeling," *IEEE Signal Processing Letters*, Vol. 6, No. 6, pp.138-140, June 1999.
13. "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification(ITU-T Rec. H.264—ISO/IEC 14496-10 AVC)," Joint Video Team(JVT) of ISO/IEC MPEG and ITU-T VCEG(ISO/IEC JTC/SG29/WG11 and ITU-T SG16 Q.6) JVT-G050.doc, 8th meeting, Geneva, Switzerland, May 2003.

The Design of Single Encoder and Decoder for Multi-view Video

Haksoo Kim and Manbae Kim

Kangwon National University
Department of Computer, Information, and Telecommunication
192-1 Hoja2-dong, Chunchon 200-701, Republic of Korea
manbae@kangwon.ac.kr

Abstract. Recently, multi-view video has been an emerging media in the 3-D field. In general, the multi-view video processing requires encoders and decoders as many as the number of cameras, and thus the processing complexity results in difficulties of practical implementation. To solve this problem, this paper considers the design of a single encoder and a single decoder for the multi-view video. At the encoder side, multi-view sequences are combined by a video mixer. Then, the mixed sequence is compressed by an H.264/AVC encoder. The decoding part is composed of a single decoder and a scheduler controlling the decoding process. The goal of the scheduler is to assign approximately identical number of decoded frames to each view sequence by estimating the decoder utilization of GOPs and subsequently applying frame skip methods. Experimental results show that identical decoder utilization is achieved for each view sequence. Finally, the performance of the proposed method is compared with a simulcast encoder in terms of bit-rate and PSNR using a RD (rate-distortion) curve.

1 Introduction

With the progress of information technologies and the demand of consumers, a variety of multimedia contents have been served to the consumers. Recently, beyond 2-D and stereoscopic video, multi-view video acquired from multiple cameras has been a newly emerging media [1,2,3,4,5], and can provide a wide view to consumers with immersive perception of natural scenes.

In general, multi-view compression systems are implemented similar to a simulcast method, where each view sequence is separately encoded by its dedicated encoder. Fecker et al. [6] proposed a simulcast method. The compressed view sequences are combined into a single bitstream that is transmitted to the client. Anthony et al. [4] developed a multi-view transmission system, where each encoded view sequence is transmitted independently. Both approaches employ encoders and decoders as many as the number of view sequences, requiring complex system design. To solve this problem, this paper presents a single encoder and decoder system for processing multi-view video. A mixed sequence is generated by combining all the view sequences and then compressed by an

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 732–741, 2006.

H.264/AVC encoder. The additional information such as the number of cameras, GOP size and encoder profile are needed prior to the encoding. At the decoder side, a single decoder is used in order to decode a compressed bitstream containing all view sequences. To assign the fair decoding to each view sequence, a scheduling scheme is utilized. The proposed decoder is designed to satisfy the constant quality of decoded sequences that is one of the requirements of the MPEG MVC [7].

The standardization of multi-view video compression is currently under way in MPEG 3DAV MVC group [8]. The reference model of a multi-view encoder has been proposed by Fraunhofer-HHI. The proposed model is based on hierarchical B-picture and adapted prediction structure. The number of decoded picture buffers (DPB) needed at encoding process is 2×GOP_length + number_of_views. For memory management, frame reordering is implemented. Starting from the first image of each view sequence, GOPs are zigzag-scanned and encoded. A main difference is that we utilize an ordinary H.264/AVC coder rather than a multi-view coder proposed in MPEG MVC. Therefore, the encoder complexity including the number of DPBs does not increase in our method.

Definitions
N : Intra period
M : Interval between I/P pictures
K : The number of view sequences (number of cameras)
V_k : kth view sequence, $k \in \{1, 2, \dots, K\}$
GOP(V_k) : GOP of V_k
GoGOP (Group of GOP) : The set of GOPs at time range $[t_i,\, t_{i+N}]$
D_k : Decoding time of V_k in GoGOP
U_k : Decoder utilization of V_k in GoGOP
S : The maximum number of decoded frames in GoGOP

This paper is organized as follows. In Section 2, the structure of a single encoder is presented along with video mixing. In Section 3, the decoding system using a single decoder and a scheduler is discussed. Experimental results are given in Section 4. The decoder performance and coding efficiency are evaluated. The conclusion and future works are given in Section 5.

2 Single Encoder with Video Mixing

In order to implement a single encoder, view sequences need to be reordered by a video mixing. Fig. 1 shows how the four view sequences are reordered by the video mixing and then compressed by an H.264/AVC encoder. K and N are 4 and 4, respectively. In Fig. 1 (a), each view sequence has an identical GOP structure (e. g., IPPPPI). The encoding order is GOP$_1$(V_1), GOP$_1$(V_2), GOP$_1$(V_3), GOP$_1$(V_4), GOP$_2$(V_1), GOP$_2$(V_2), ... as illustrated in Fig. 1 (b). The multi-view sequences are encoded in GOP units. The ith group of GOPs, GoGOP$_i$ consists of GOP$_i$(V_1), GOP$_i$(V_2), GOP$_i$(V_3), and GOP$_i$(V_4). The video mixing is designed in a manner that the multi-view video can be encoded by a conventional H.264/AVC encoder, thus maintaining the identical encoder complexity. The video mixing can be applied to baseline and main profiles of H.264/AVC. The main profile includes B picture in addition to I and P pictures of the baseline profile [9].

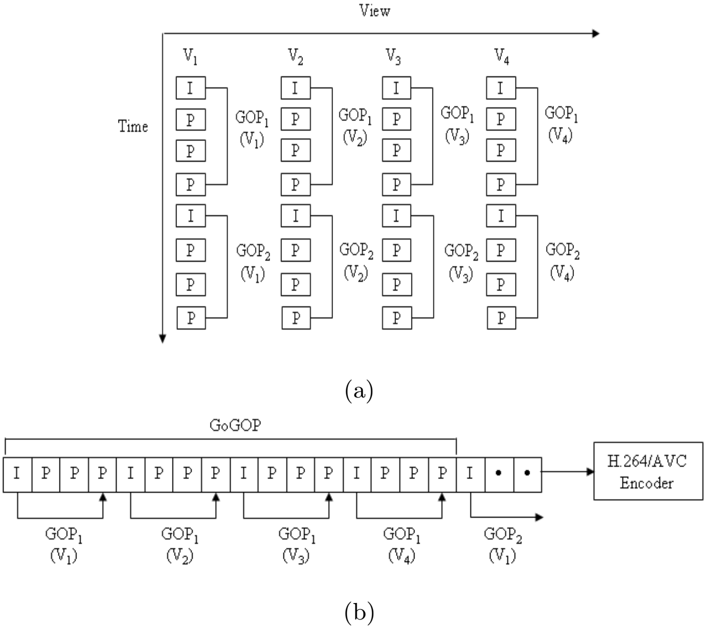


Fig. 1. shows how view sequences are mixed and compressed. (a) Four view sequences, and (b) mixed sequences compressed by an H.264/AVC encoder.

3 Decoding Multi-view Video

This section presents a multi-view decoding system that is composed of an H.264/AVC decoder and a scheduler controlling the decoder. Even though different scheduling schemes are carried out for the baseline and main profiles, they are designed in a manner that each encoded view sequence maintains the consistent quality of decoded sequences (e. g., the number of decoded frames). The scheduler is composed of the three parts: the computation of decoder utilization of each view sequence, the view priority decision, and the frame skip. Since the number of frames to be decoded are limited (e. g., 30 fps), the scheduler effectively controls the number of decoded frames for each view sequence. Further, in order to reduce the degradation of visual image quality caused by the frame skip process, an effective scheme for selecting discarded frames based upon a cost function is presented. Finally, we discuss the frame skip schemes for the baseline and main profiles.

The block diagram of the proposed multi-view decoder is illustrated in Fig. 2. The compressed bitstream of K view sequences is separated by the stream splitter and each compressed data of each view is stored at buffer(V_k) ($k = 1, \ldots, K$). The scheduler manages the decoding of the stored buffer data in GOP units. For a GOP of V_k, the decoding time D_k is computed. Then the decoder utilization U_k is computed by

$$U_k = D_k / \sum_{k=1}^{K} D_k \tag{1}$$

where the range of U_k is [0, 1]

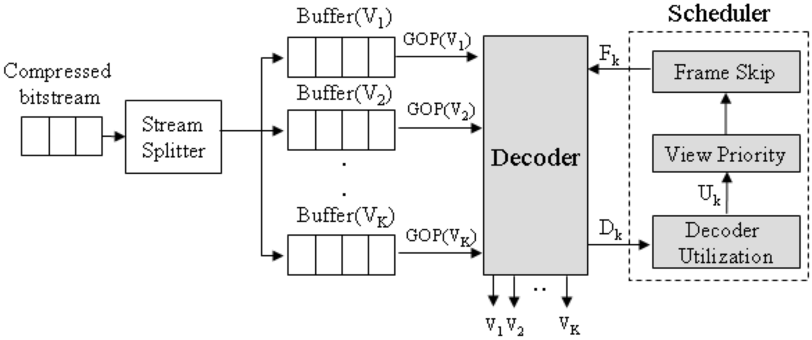


Fig. 2. Block diagram of the proposed multi-view decoding system

The view priority adaptively varies every GoGOP according to the U_k of a current GOP. A view sequence with low U_k is assigned a higher priority and thus decoded earlier in the next GOP. The frame skip is mainly composed of skipped frame decision and decoded frame selection as shown in Fig. 3. The former decides the number of discarded frames among N GOP frames. For every V_k, the number of discarded frames l_k is estimated from U_k. According to the l_k, the decoded frame selection chooses frames to be decoded based upon a skip mode that consists of sequential skip and non-sequential skip. For the main profile of H.264/AVC, both skip modes can be used. Meanwhile, only the sequential skip mode is used for the baseline profile. Both modes utilize a cost function that is related to visual image quality [10,11]. Finally, GOP frames to be decoded are determined.

Suppose that if lth frame F_l in a GOP(V_k) is discarded, successive F_{l+1}, F_{l+2}, ..., F_N are also discarded. Then, the number of decoded frames is $(l - 1)$. For all GOPs, the total decoded frames in a GoGOP is computed by

$$(l_1 - 1) + (l_2 - 1) + \cdots + (l_K - 1) = \sum_{k=1}^{K} l_k - K \tag{2}$$

where l_k is the first frame to be discarded in the GOP(V_k).

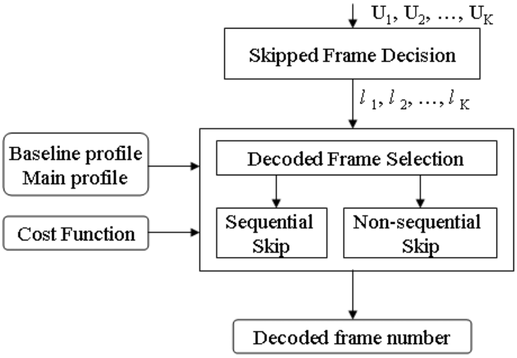


Fig. 3. Block diagram of the proposed frame skip

The maximum number of decoded frames in a GoGOP is $K \times N$. However, since typical computing power can not satisfy such a requirement, the number of actually decoded frames, S is less than or equal to $K \times N$. Then, Eq. (2) becomes

$$\sum_{k=1}^{K} l_k - K = S \tag{3}$$

where the range of S is at $[0,\ K \times N]$.

We utilize Eq. (2) to find $l_1, l_2, \ldots, l_K$ to satisfy Eq. (3). Given U_k, l_k is estimated by using an inverse relation. If $U_k = [0,\ 1]$ is inversely mapped to $l_k = [S, 1]$, this is expressed by

$$\hat{l_k} = -(S-1)U_k + S \tag{4}$$

where $\hat{l_k}$ is the estimate.

Summing over K views, we have

$$\sum_{k=1}^{K} \hat{l_k} = \sum_{k=1}^{K} [-(S-1)U_k + S] = S(K-1) + 1 \tag{5}$$

where $\sum_{k=1}^{K} U_k = 1$.

From Eqs. (3) and (5), the first frame number to be discarded for GOP(V_k) is computed by

$$l_k^* = \lfloor (S+K) \frac{\hat{l_k}}{\sum_{k=1}^{K} \hat{l_k}} \rfloor \tag{6}$$

where $\lfloor\ \rfloor$ is a floor function. The number of decoded frames is $l_k^* - 1$.

The next procedure is to determine which frames are decoded. A simple approach is to decode all consecutive frames before the first discarded frame.

The other is to discard non-consecutive frames, thus achieving better visual image quality.

In order to provide a measure of the playback discontinuity, a *cost function*, $\phi(S)$ is defined. It takes two aspects of playback discontinuity into account: the length of a sequence of consecutive discarded frames and the distance between two adjacent but non-consecutive discarded frames. The cost function $\phi(S)$ assigns a cost c_i to a discarded frame F depending on whether it belongs to a sequence of consecutive discarded frames or not. If frame F belongs to a sequence of consecutive discarded frames, then the cost c_i is defined to be m_i, if the frame is the m_ith consecutively discarded frames in the sequence. Otherwise, the cost function c_i is defined based on the distance d_i to the previous discarded frame and given by the formula $c_i = 1+1/\sqrt{d_i}$. Therefore, the total cost function is the sum of all the costs of any discarded frames in the sequence.

Since the baseline profile of H.264/AVC defines I and P pictures only, a sequential skip mode is used. In case of the GOP structure of I1 P1 P2 P3 P4 P5 I2 P6 P7 P8, if I1 is discarded, next P1, ... , P5 are also discarded. Therefore, If Pi is discarded, all successive frames in the same GOP are discarded. Therefore, if three frames in the current GOP and two frames in the next GOP are discarded, the decoded frames are I1, P1, P2 and I2, P6.

Unlike the baseline profile, two skip modes can be utilized for the main profile. However, the non-sequential mode gives less cost function than the sequential one. The selection of decoded frames depends upon picture types of I, P and B. Table 1 shows some examples of the frame selection with varying GOP structures. The selection order indicates the degree of importance for consistent visual image quality. For the main profile with $N = 9$ and $M = 3$, the selection order is I P1 P2 B1 B3 B5 B2 B4 B6. If the total decoded frames are six, I P1 P2 B1 B3 and B5 are decoded. The selection order can also be found in other two examples.

Table 1. The non-sequential skip mode and the selection order

Main profile (N=9, M=3)	I	B1	B2	P1	B3	B4	P2	B5	B6	
Selection order	1	4	7	2	5	8	3	6	9	
Main profile (N=8, M=4)	I	B1	B2	B3	P1	B4	B5	B6		
Selection order	1	3	5	7	2	4	6	8		
Main profile (N=10, M=5)	I	B1	B2	B3	B4	P1	B5	B6	B7	B8
Selection order	1	3	5	7	9	2	4	6	8	10

4 Experimental Results

We have performed various experiments in order to examine the performance of the proposed single encoder and decoder design. Test multi-view sequences are *aquarium*, *flamenco*, *objects*, and *race* provided by KDDI, Japan [8]. The image size of the test sequences is 320 × 240. We used H.264/AVC JM (Joint

Model) 7.6 reference software. Bit rates are 128, 256, and 512 Kbps, and QP (quantization parameter) is fixed to be 28. Intra Decoded Refresh (IDR) is set to zero.

As given in Table 2, mixed test sequences for baseline and main profiles were produced with various combinations of (K, N, M) and encoded by H.264/AVC. b21.yuv with $(K, N) = (2, 12)$ is made by combining *flamenco1.yuv* and *aquarium1.yuv*. The total number of frames N_T is 300 and the encoded file is b21.264. The reason why different sequences are mixed is that we tried to examine the effect of inter-coding modes of H.264/AVC on decoded frames. m21.yuv with $(K, N, M) = (2, 12, 3)$ was made from *flamenco1.yuv* and *aquarium1.yuv* and its compressed bistream is m21.264. The total frames are 240. Other test sequences were produced in a similar manner. Fig. 4 shows the variation of decoder utilization of the baseline profile sequences. In case of b41.264 with four views, the decoder utilization is approximately 0.25. Similar results are also observed for b81.264, where the mean decoder utilization is approximately 0.125. Fig. 5 shows the decoder utilization of two main profile sequences, m21.264 and m43.264. The graphs show similar performance with the baseline profile sequences. The variation of cost functions for two main profile sequences, m41.264 and m81.264 is observed in Fig. 6. As predicted, the cost function decreases linearly as S increases. The total number of frames in a GoGOP for m41.264 and m81.264 is 48 and 96, respectively. At $S = 40$ and 80, the total costs are close to zero.

Table 2. Test sequences of baseline and main profiles. N_T = total frames.

Baseline profile	K	N	N_T	Encoded file
b11.yuv	1	4	100	b11.264
b12.yuv	1	12	300	b12.264
b21.yuv	2	12	300	b21.264
b22.yuv	2	24	300	b22.264
b41.yuv	4	12	400	b41.264
b42.yuv	4	24	400	b42.264
b81.yuv	8	12	500	b81.264
b82.yuv	8	24	500	b82.264

Main profile	K	N	M	N_T	Encoded file
m11.yuv	1	4	1	80	m11.264
m12.yuv	1	12	2	120	m12.264
m21.yuv	2	12	3	240	m21.264
m22.yuv	2	12	2	240	m22.264
m23.yuv	2	12	6	240	m23.264
m41.yuv	4	12	6	480	m41.264
m42.yuv	4	12	4	480	m42.264
m43.yuv	4	16	8	512	m43.264
m81.yuv	8	12	3	576	m81.264
m82.yuv	8	24	6	576	m82.264

We compared the RD (rate-distortion) performance of the proposed method with a simulcast method as given in Fig. 7. The GOP structure is IBBP... and N is 12. For *flamenco1*, the simulcast method outperforms ours by PSNR of 2 dB on average. On the contrary, our method outperforms the simulcast one by 8 dB for *race1*. For *objects1* with eight view sequences, two methods show the similar performance. Experimental results show that the performance depends upon image motions contained in multi-view video.

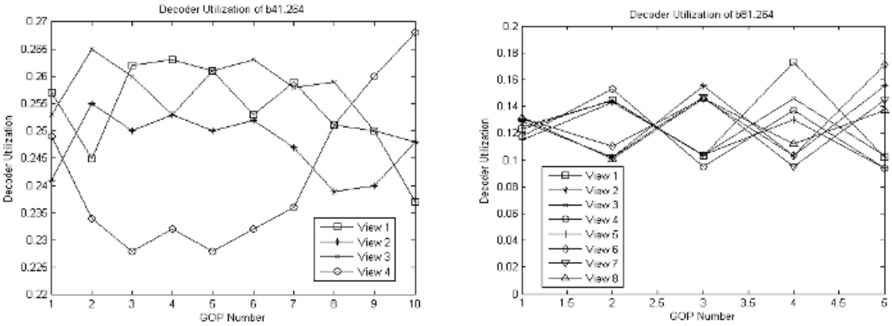


Fig. 4. The variation of decoder utilization for baseline profile sequences

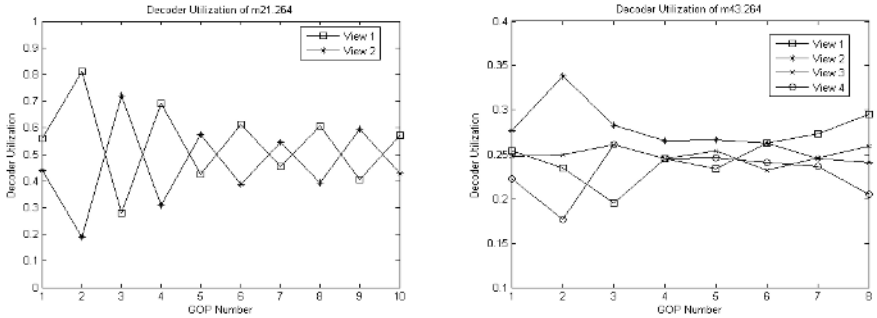


Fig. 5. The variation of decoder utilization for main profile sequences

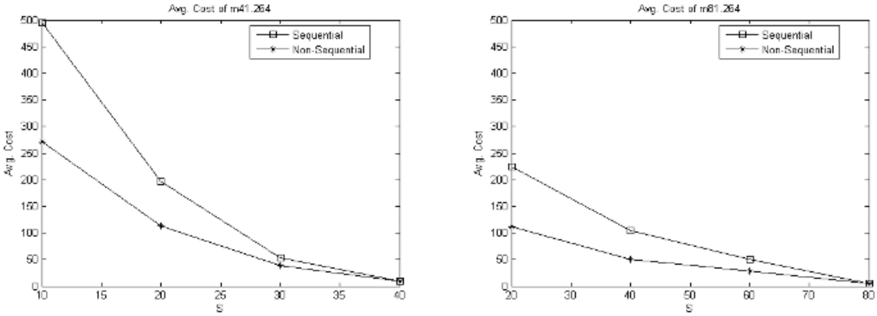


Fig. 6. The variation of cost functions for main profile sequences with respect to S

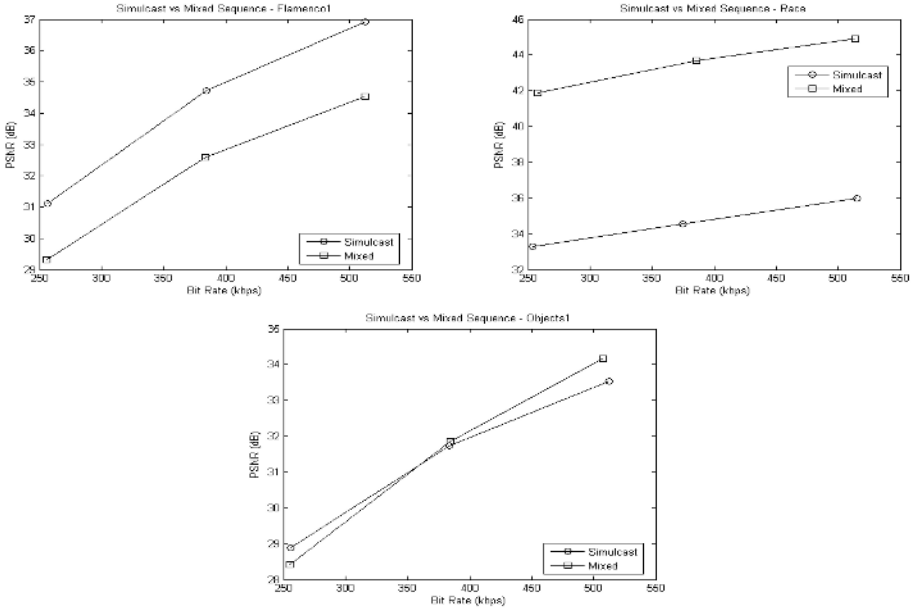


Fig. 7. Performance comparison of simulcast sequence and video-mixed sequence

5 Conclusion and Future Works

In this paper, we presented a single encoder and decoder design for multi-view video processing compared with the simulcast approach requiring encoders and decoders as many as the number of camera views. Further, conventional H.264/AVC encoder and decoder were employed unlike MPEG MVC multi-view coder. To carry out such design, a video mixing reordering view sequences was proposed. At the decoder side, an appropriate scheduling scheme as well as frame skip modes was proposed. For the consistent quality of decoded view sequences, decoder utilization was utilized to decide the frames to be decoded. The cost function was presented for baseline and main profiles of H.264/AVC for controlling visual image quality. Experimental results performed on various multi-view test sequences validate that each view sequence is fairly decoded. Finally, the coding efficiency verified by RD curves showed that the proposed method achieves similar performance compared with the simulcast method.

For future works, we are going to study the expansion of our method to multiple decoders since this could produce better performance. Secondly, clients might ask for particular views rather than the demand of all the view sequences. Then, our system needs to be adapted to this application. One approach will be the application of DIA (Digital Item Adaptation) of MPEG-21 Multimedia Framework.

Acknowledgment

Authors would like to thank anonymous reviewers for their valuable comments. This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment)(RBRC: IITA-2006-(C1090-0502-0022)).

References

1. E. Cooke, I. Feldman, P. Kauff, and O. Schreer, ”A modular approach to virtual view creation for a scalable immersive teleconferencing”, IEEE International Conf. on Image Processing, 2003.
2. W. Matusik and H. Pfister, ”3D TV: a scalable system for real-time acquisition, transmission and autostereoscopic display of dynamic scenes”, Proc. of ACM SIGGRAPH, 2004.
3. S. Lee, K. Lee, C. Han, J. Yoo, M. Kim and J. Kim, ”View-switchable stereoscopic HD video over IP networks for next generation broadcasting”, SPIE Photonic East, Vol. 6016, Oct. 2005.
4. A. Vetro, W. Matusik, H. Pfister, J. Xin, ”Coding approaches for end-to-end 3D TV systems”, Pciture Coding Symposium, Dec. 2004.
5. Q. Zhang, W. Zhu and Y-Q Zhang, ”Resource allocation for multimedia streaming over the Internet”, IEEE Trans. on Multimedia, Vol. 3, No. 3, Sep. 2001.
6. Ulrich Fecker and Andre Kaup, ”Transposed Picture Ordering for Dynamic Light Field Coding”, ISO/IEC JTC1/SC29/WG11, N10929, Redmond USA, July 2004.
7. MPEG/ISO/IEC JTC1/SC29/WG11, ”Requirements on Multi-view Vdeo Coding”, N6501, Redmond, USA, July 2004.
8. MPEG/ISO/IEC JTC1/SC29/WG11, ”Mutiview Coding using AVC”, M12945, Bangkok, Thailand, Jan. 2006.
9. I. Richardson, *H.264 and MPEG-4 video compression*, Wiley, 2003.
10. K. Chebrolu and R. Rao, ”Selective frame discard for interactive video”, IEEE International Conference on Communications, Vol. 7, pp. 4097-4102, June 2004.
11. Z.-L Zhang, S. Nelakuditi, R. Agarwal, and R. Tsang, ”Efficient selective frame discard algorithms for stored video delivery across resource constrained networks”, IEEE INFOCOM, 1999.

Vector Quantization in SPIHT Image Codec

Rafi Mohammad and Christopher F. Barnes

School of Electrical and Computer Engineering, Georgia Institute of Technology,
Atlanta, GA 30332-0250, USA
rafi@ece.gatech.edu, chris.barnes@gatech.edu

Abstract. The image coding[1] algorithm "Set Partitioning in Hierarchical Trees (SPIHT)" introduced by Said and Pearlman achieved an excellent rate-distortion performance by an efficient ordering of wavelet coefficients into subsets and bit plane quantization of significant coefficients. We observe that there is high correlation among the significant coefficients in each SPIHT pass. Hence, in this paper we propose trained scalar-vector quantization (depending on a boundary threshold) of significant coefficients to exploit correlation. In each pass, the decoder reconstructs coefficients with scalar or vector quantized values rather than with bit plane quantized values. Our coding method outperforms the scalar SPIHT coding in the high bit-rate region for standard test images.

1 Introduction

All natural images consist of both trends (low frequency content) and anomalies (high frequency content) and wavelet transform is well suited for capturing these phenomena in a small fraction of total coefficients. Due to this excellent energy compaction property, wavelet transform based image compression has grown popular over the last decade [1], [2], [3], [4]. Also, reconstructed images are free of blocking artifacts common to DCT compressed images. Shapiro's Embedded Zerotree Wavelet (EZW) coder achieved excellent rate-distortion performance by using a novel zerotree data structure [2], successive approximation quantization and arithmetic coding. EZW produces an embedded bit stream that is suitable for remote browsing applications. Said and Pearlman introduced the SPIHT embedded coder [4] as an improvement of EZW, and its performance surpassed the performance of all existing image coding algorithms. Mukherjee and Mitra [5] introduced vector SPIHT (VSPIHT) for image coding by forming blocks of wavelet coefficients before zerotree coding. VSPIHT algorithm for image compression achieved modest improvement over scalar SPIHT [4] algorithm.

In this paper we propose a vector quantization of significant coefficients of each pass of SPIHT below a boundary threshold and scalar quantization of significant coefficients above the same threshold. The resulting indices are sent to the decoder for decoding the compressed image. Here, for simplicity we have not performed any further compression of indices through entropy coding. Unlike

[1] The terms *coding* and *compression* are used synonymously in this paper.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 742–750, 2006.

in [5], we form blocks of significant coefficients after they are found using zerotree prediction.

Section 2 explains briefly SPIHT image coding algorithm, and the analysis of significant data using correlation. In Section 3, we explain how to utilize the correlation among coefficients to efficiently compress the significant coefficients using a combination of scalar and vector quantization based on the Lloyd algorithm [6]. Section 4 gives the coding results of standard test images using this scalar and vector quantized SPIHT (SQ-VQ SPIHT) algorithm. Section 5 concludes the paper.

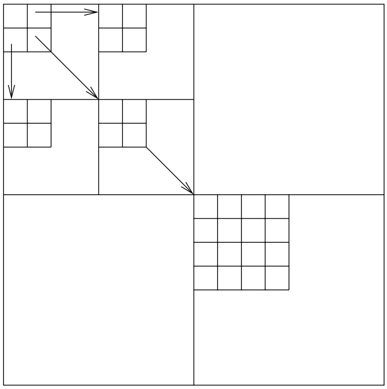


Fig. 1. Spatial Orientation Tree

2 Analysis of Significant Coefficients in SPIHT

In a hierarchical subband decomposition such as wavelets there exists a relationship among coefficients across scales in the form of a tree known as the Spatial Orientation Tree (SOT) [4]. SOT consists of a root node and all of its descendants as shown in Fig. 1 for a 3 level dyadic subband decomposition. SPIHT uses three lists to encode and decode significant information: List of insignificant pixels (LIP), List of insignificant sets (LIS) and List of significant pixels (LSP). In each pass coefficients are classified into these lists using a significance test [4].

Every SPIHT pass consists of sorting, refinement and quantization update steps and coefficients are compared to a threshold ($T = 2^n$) for significance. In the initial pass, T is set as 2^{n_0}, where n_0 is chosen such that maximum coefficient $c_{\max} \in [2^{n_0},\ 2^{n_0+1})$. LIP is initialized with the elements of the highest level low-low frequency subband and LIS is initialized with SOT root nodes. During the sorting step of a pass, LIP elements are tested for significance and moved to LSP, if found significant. Also, the sets in LIS are tested for significance in the same step and partitioned into subsets. In the refinement step, the elements in the LSP are refined by adding n^{th} most significant bit (MSB) to the encoder bit stream in the following pass. n is decreased by 1 before the next pass in the quantization update step. As the coefficients are successively refined using threshold as a power of two, SPIHT encoding can be considered to be Bit Plane Coding (BPC) [7]. BPC is similar to the binary representation of real numbers, with each binary digit added to the right, more precision is added.

Correlation between coefficients can be quantified using an autocorrelation function (acf) [8]. The autocorrelation function gives the closeness or similarity of two samples as a function of their space or time separation (k) on average. Variance-normalized acf is defined as

$$\rho_k = \frac{R_{cc}(k)}{R_{cc}(0)} = \frac{1}{N_c \sigma_c^2} \sum_{n=0}^{N_c-|k|-1} c(n)c(n+|k|) \tag{1}$$

$$R_{cc}(0) = \sigma_c^2 \tag{2}$$

where N_c = Number of coefficients in a particular pass i.e. coefficients having magnitude in the interval $[T\ 2T)$, σ_c^2 = variance of coefficients and $R_{cc}(k)$ is the autocorrelation function. We observe that there is a high correlation among coefficients resulting after each sorting step and Fig.2 gives a variation of ρ_k with k for various intervals for a standard test image, Barbara. Note that the coefficients in each pass have absolute values and their signs are encoded separately.

Similar correlation is found among the coefficients of each sorting step in all the natural images tested. From Fig.2, it can be seen that absolute coefficients with magnitude less than 1 have $\rho_k > 0.95$ for $k = 1, 2, \cdots, 10$.

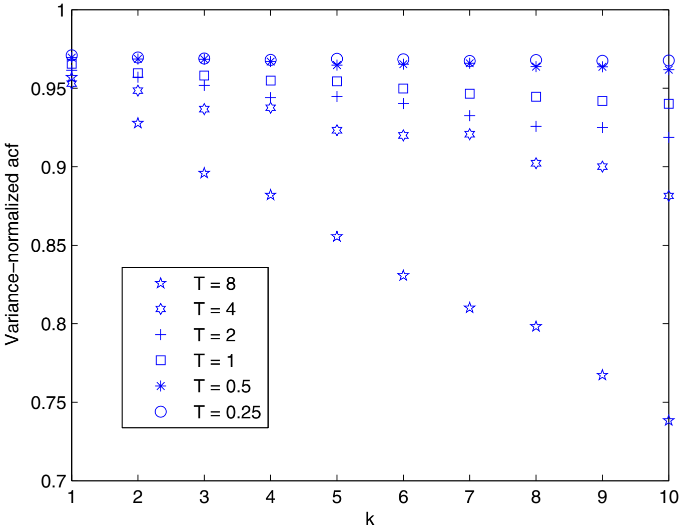


Fig. 2. Variation of normalized acf with k, (T represents the interval $[T, 2T)$)

3 Scalar-Vector Quantization of Significant Coefficients

Shannon's rate-distortion theory tells that better compression can be achieved by coding vectors rather than scalars [9]. All natural images have a high probability of having insignificant coefficients and these can be coarsely quantized or discarded [10]. Initial few passes of SPIHT for all natural images result in significant coefficients of high magnitude, hence these coefficients need to be scalar quantized for finer classification. Therefore, we choose to vector quantize the coefficients below and scalar quantize the coefficients above a boundary threshold T_{sq-vq}. The value of T_{sq-vq} has to be always less than the initial quantization level $T = 2^{n_0}$ so that coefficients of few initial passes can be scalar quantized.

In a fixed rate vector quantizer of codebook size N and block length m, every code vector, y_i, is represented by the same number of bits ($\log_2 N$) and the code rate of this VQ is $\log_2 N/m$ bits per vector component. The goal of optimal vector quantizer is to design a codebook $Y = (y_i, i = 1, 2, \cdots N)$ such that the expected distortion between an input vector x and its reproduction vector $\hat{x}$ is minimized. The LBG algorithm [11], which is an extension of the Lloyd algorithm for scalar quantizers was developed to produce such a codebook for vector quantizer. For $m = 1$, vector quantization (VQ) becomes non-uniform scalar quantization (SQ).

SQ-VQ SPIHT image encoding algorithm is given in Fig.3 and decoding algorithm will be similar [4] to the encoding algorithm. The refinement step of SPIHT algorithm is replaced by the scalar-vector quantization step. SQ-VQ SPIHT decoder reconstructs the coefficients simply as a table look up operation for scalar and vector quantizers.

For a p pass image coding, the total number of bits spent in SPIHT

$$n_{b_1} = \sum_{i=1}^{p-1} n_{e_i} + n_{b_{sort}} \tag{3}$$

The total number of bits spent in the scalar-vector quantized SPIHT in p passes

$$n_{b_2} = \sum_{i=1}^{p_{sq}} n_{e_i} \log_2 N_1 + \sum_{i=p_{sq}+1}^{p} n_{e_i} \frac{\log_2 N_2}{m} + n_{b_{sort}} \tag{4}$$

Here n_{e_i} is the number of elements in a sorting step i, p_{sq} is the number of passes in which coefficients are scalar quantized, N_1 scalar quantizer codebook size, N_2 vector quantizer codebook size, $n_{b_{sort}}$ is the total number of sorting step bits in p passes. Here, sorting step coefficients are quantized in the same pass, whereas in SPIHT current sorting step coefficients are refined in the refinement step of the following pass.

4 Implementation and Results

With the above background, we present the actual implementation details and coding results for standard gray scale images used by the image coding research

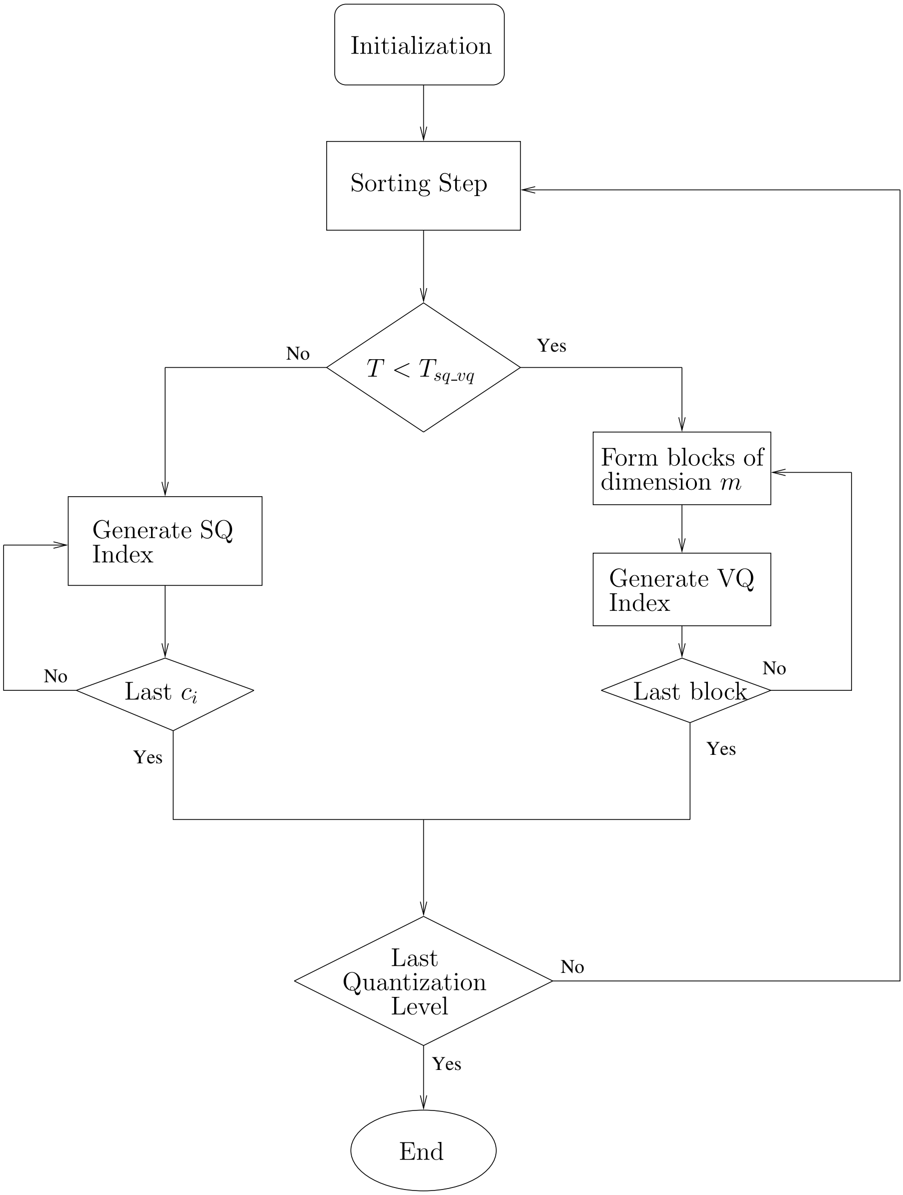


Fig. 3. SQ-VQ SPIHT Image encoder

community. Each image of N_{tr} number of natural images is wavelet transformed using bi-orthogonal filters for 5 levels. After subtracting the mean from the highest level low-low subband, the coefficients are grouped into several classes. Each class has absolute value of coefficients in $[2^{n_0-n}\ 2^{n_0-n+1})$ where n_0 is

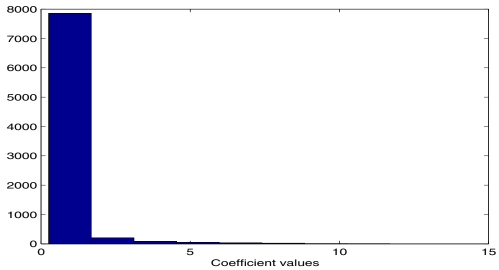


Fig. 4. Histogram plot of significant wavelet coefficients for Barbara image

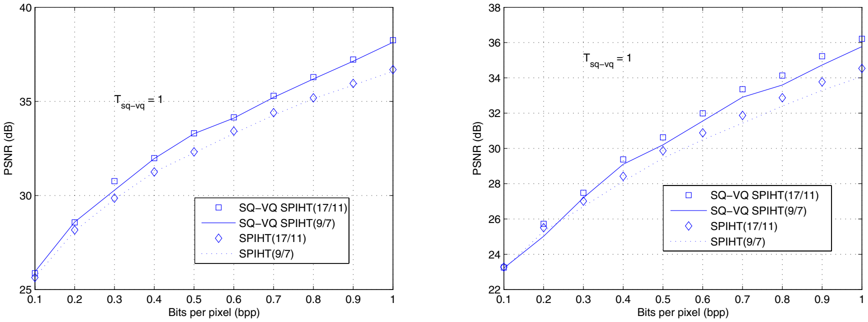


Fig. 5. Rate-distortion curves for (a) Boat and (b) Barbara images using 17/11 and 9/7 bi-orthogonal filters

chosen as per the maximum value of coefficients and n takes values $0, 1, \cdots, p_{\max}$. Here, $p_{\max}$ is chosen as per the highest bit rate requirement.

Fig.4 shows the histogram plot of significant coefficients resulting from all SPIHT passes at 0.1 bits per pixel (bpp) for Barbara image. We can see from the histogram that most of the coefficients are close to zero in magnitude and these coefficients need to be vector quantized by forming blocks for efficient compression. A few high magnitude coefficients have to be scalar quantized for finer classification. We find the distribution of significant coefficients similar to this histogram for all natural images tested. Selecting a suitable boundary threshold, T_{sq-vq}, between scalar and vector quantizers is an important step. It is found that the threshold T_{sq-vq} is highly image dependent and its optimal value will result in good compression performance.

A trained scalar codebook of size N_1 is formed for each class of coefficients having a magnitude greater than T_{sq-vq}. For vector quantization, training data of block length m is formed from each class of coefficients having magnitude below T_{sq-vq}. Then, these vectors of each class are quantized using the LBG algorithm [11]. Thus, we have scalar codebooks for each class of coefficients above T_{sq_vq} and vector codebooks below T_{sq-vq}.

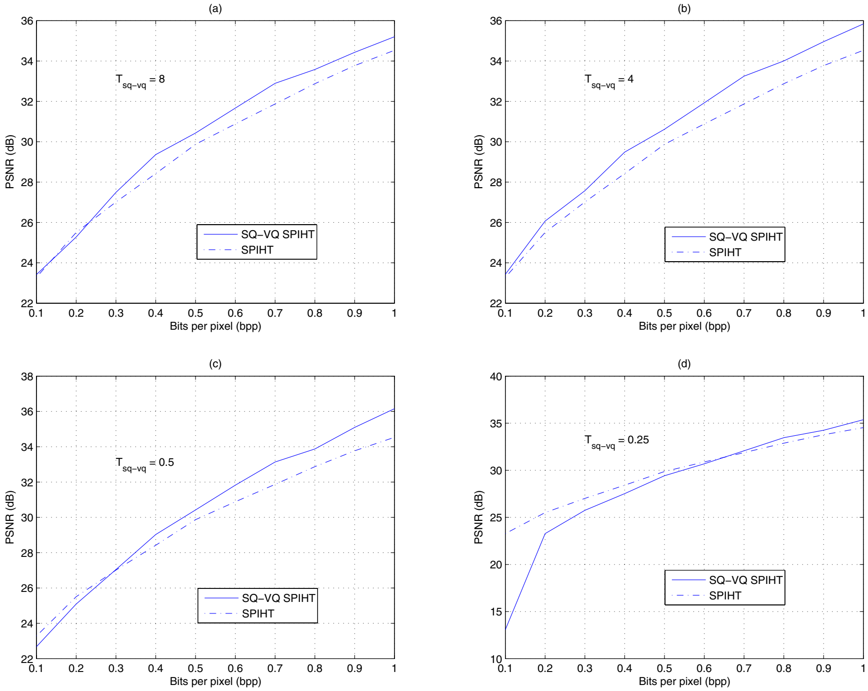
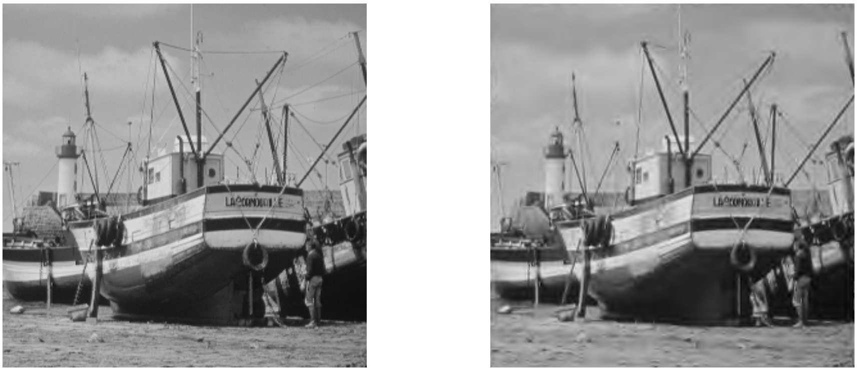


Fig. 6. Rate-distortion curves at various boundary thresholds(T_{sq-vq}) for Barbara image

To encode an image in SQ-VQ SPIHT, it is first wavelet transformed using bi-orthogonal filters for 5 levels and then the mean is subtracted from its top level low-low band. After the sorting pass [4], coefficients in the interval $[T\ 2T)$, where $T \geq T_{sq-vq}$, are scalar quantized with a codebook of size N_1 and the coefficients in the interval $[T\ 2T)$, where $T < T_{sq-vq}$, are vector quantized with a codebook of size N_2. The scalar and vector indices are sent to the decoder using $\log_2 N_1$ and $\log_2 N_2/m$ bits respectively for each coefficient. For simplicity, the scalar and vector quantized indices are not entropy-coded. We have used $N_{tr} = 230$ training images , code book sizes $N_1 = 16, N_2 = 256$ and block length $m = 4$ in all our coding experiments. Also, in our experiments we have not included test images while forming the training data.

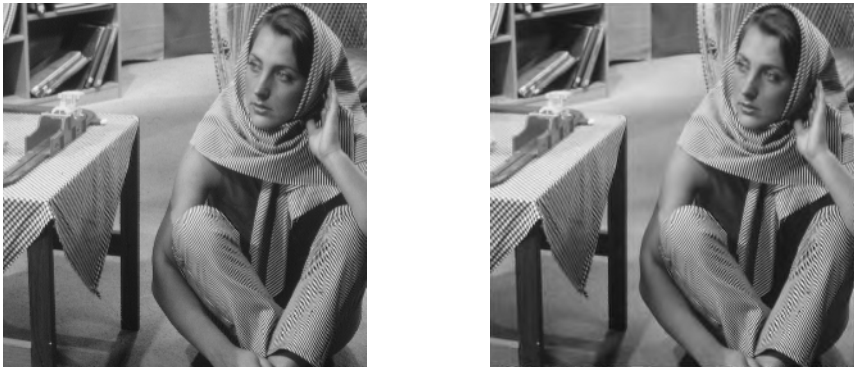
From the rate-distortion curves given in Fig.5 for Boat and Barbara images, it can be seen that SQ-VQ SPIHT performs better in the high bit region compared to low bit rate region. Also, there is negligble effect on rate-distortion performance with the type of bi-orthogonal filter (17/11 or 9/7) used. To see the effect of the boundary threshold T_{sq-vq} on rate-distortion performance, we have performed coding experiments at various T_{sq-vq} values for Barbara image, as shown in Fig.6(a), (b), (c) and (d) using 17/11 bi-orthogonal filters. It can be seen from these figures that as T_{sq-vq} is reduced, mean square error of coded



(a) Original Image

(b) SQ-VQ SPIHT at 0.2 bpp (PSNR = 28.57 dB)

Fig. 7. Visual Quality results for the Boat Image



(a) Original Image

(b) SQ-VQ SPIHT at 0.65 bpp (PSNR = 32.69 dB)

Fig. 8. Visual Quality results for the Barbara Image

images increases, resulting in poor PSNR values. At low T_{sq-vq} values, a large number of significant coefficients need to be scalar quantized compared to the high T_{sq-vq} values. This will incur high index bit cost resulting in poor rate-distortion performance. At high T_{sq-vq} value, a few significant coefficients need to be scalar quantized and the rest are vector quantized resulting in good rate-distortion performance (as index bit cost per coefficient is less in VQ than in SQ) as shown in Fig.6(a) and (b). Particularly from Fig.6(b), it can be seen that the boundary threshold $T_{sq-vq} = 4$ results in the best rate distortion performance for Barbara image. Fig.7 and 8 show visual quality of boat and Barbara images coded at 0.2 bpp and 0.65 bpp respectively with SQ-VQ SPIHT algorithm using $T_{sq-vq} = 1$ and 17/11 bi-orthogonal filters.

The SQ-VQ SPIHT encoding complexity is higher than that of SPIHT due to exhaustive search vector quantization and non uniform scalar quantization. But the decoding complexity of the same is less than that of SPIHT due to table look up operation. The non-uniform SQ and VQ complexity can be reduced significantly by using residual quantizers (scalar and vector) [12] in SQ-VQ SPIHT.

5 Conclusion

We have presented scalar and vector quantization of significant coefficients in SPIHT image codec. SQ-VQ SPIHT performs better than SPIHT coding algorithm in the high bit region compared to low bit rate region. We found that choosing a suitable boundary threshold results in good rate-distortion performance. In future, we need to find an automatic method to calculate image specific boundary threshold. Also SQ-VQ SPIHT performance can be improved further by employing variable rate residual vector quantization instead of exhaustive search vector quantization as used in this work.

References

1. M.Antonini, M.Barlaud, P.Mathieu, and I.Daubechies, “Image coding using wavelet transform,” *IEEE Trans. Image Processing*, vol. 1, pp. 205–220, Apr. 1992.
2. Jerome M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
3. E.A.B Da Silva, D.G. Sampson, and M.Ghanbari, “A successive approximation vector quantizer for wavelet transform image coding,” *IEEE Trans. Image Processing*, vol. 5, pp. 299–310, Feb. 1996.
4. Amir Said and William Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.
5. Debargha Mukherjee and Sanjit K. Mitra, “Vector spiht for embedded wavelet video and image coding,” *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 13, pp. 231–246, Mar. 2003.
6. Allen Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.
7. Brian A. Banister and Thomas R. Fischer, “Quantization performance in spiht and related wavelet image compression algorithms,” *IEEE Sig. Proc. Letters*, vol. 6, pp. 97–99, May. 1999.
8. N. S. Jayant and Peter Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984.
9. Robert M. Gray, “Vector quantization,” *IEEE ASSP Mag.*, vol. 1, pp. 4–29, Apr. 1984.
10. Al Bovik, *Hand book of Image and Video Processing*, Academic Press, San Diego, USA, 2000.
11. A. Buzo Y.Linde and R.M. Gray, “An algorithm for vector quantizer design,” *IEEE Trans. Commun.*, pp. 84–95, Jan.1980.
12. Christopher F. Barnes, Syed A. Rizvi, and Nasser M. Nasrabadi, “Advances in residual vector quantization: A review,” *IEEE Trans. Image Processing*, vol. 5, pp. 226–262, Feb. 1996.

Objective Image Quality Evaluation for JPEG, JPEG 2000, and Vidware Vision^{TM}

Chung-Hao Chen, Yi Yao, David L. Page, Besma Abidi, Andreas Koschan,
and Mongi Abidi

The University of Tennessee
Imaging, Robotics and Intelligent System Laboratory
Knoxville, Tennessee 37996-2100, USA
{cchen10, yyao1, dpage, besma, akoschan, abidi}@utk.edu

Abstract. In this paper, three compression methods, JPEG, JPEG 2000, and Vidware Vision^{TM} are evaluated by different full- and no-reference objective image quality measures including Peak-Signal-to-Noise-Ratio (PSNR), structural similarity (SSIM), and Tenengrad. In the meantime, we also propose an image sharpness measure, non-separable rational function based Tenengrad (NSRT$_2$), to address whether the compression method is appropriate to be used in a machine recognition application. Based on our experimental results Vidware Vision^{TM} is more robust to changes in compression ratio and presents gradually degraded performance at a considerably slower speed thus outperforming JPEG and JPEG 2000 when the compression ratio is smaller than 0.7%. Furthermore, the effectiveness of our proposed measurement, NSRT$_2$, is also validated via experiments and performance comparisons with other objective image quality measures.

Keywords: JPEG, JPEG 2000, Vidware Vision^{TM}, image quality, mage sharpness, and image compression.

1 Introduction

Image compression [1], [2], [3] is an essential component of image retrieval [5], recognition [6], and internet applications [7]. The goal of compression is to minimize the size of the data being broadcast or stored, thus minimizing transmission time and storage space while still maintaining a desired quality compared to the original image.

One of the most popular lossy compression approaches for still images is JPEG. JPEG [2] stands for Joint Photographic Experts Group, the name of the committee that developed the standard. JPEG compression divides the input image into 8 by 8 pixel blocks and calculates the discrete cosine transform (DCT) of each block. A quantizer rounds off these DCT coefficients according to a pre-defined quantization matrix. This quantization step produces the "lossy" nature of JPEG. Afterwards, JPEG applies a variable length code to these quantized coefficients.

JPEG 2000 [1] is a wavelet-based image compression standard that was also created by the Joint Photographic Experts Group with intention to outperform their

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 751 – 760, 2006.

original JPEG standard. JPEG 2000 is based on the idea that coefficients of a transform which decorrelates pixels of an image could be coded more effectively than the original pixels. If functions of the transform, which is the wavelets transform in JPEG 2000, translate most of the important visual information into a small number of coefficients then the remaining coefficients could be quantized/normalized coarsely or even truncated to zero without introducing noticeable distortions. Compared with JPEG, JPEG 2000 [11] not only improves the quality of decompressed images but also supports much lower compression ratios.

Recently, Vidware Incorporated developed a product line called Vidware Vision^{TM} [3], [4]. It is divided into three separate categories: Still Image (as a replacement for JPEG), Full Frame video (as a replacement for M-JPEG), and Full Motion video (an H.264 compliant CODEC). Vidware Vision^{TM} Still Image is an integer-based encoding algorithm and varies the size of macro-blocks according to the shape and contents of the image, unlike JPEG where the block size is fixed. Therefore, this variable macro-blocking produces reduced blocky artifacts and increases image fidelity.

In terms of full-reference image quality measure, PSNR is the most widely used full-reference quality measure. It is appealing because of its simple computation and clear physical meaning [9], [10]. Nevertheless, it is not closely matched to perceived visual quality [8], [9], [10]. Therefore, SSIM [12] is selected for its improved representation of visual perception. The SSIM measure compares local patterns of pixel intensities that have been normalized and hence are invariant to luminance and contrast. This method could be seen as complementary to the traditional PSNR approach.

A successful recognition requires sufficient local details to differentiate various patterns and the ability to preserve these local details. These details translate into the sharpness of the decompressed images and interpreted by image sharpness measures. Sharpness measures have been traditionally divided into 5 categories [13]: gradient based, variance based, correlation based, histogram based, and frequency domain based methods. Image noise level and artifacts, such as blocking effects, vary with respect to different compression methods as well as various compression ratios. Conventional sharpness measures are not feasible because they are unable to differentiate variations caused by actual image edges from those induced by image noise and artifacts. To avoid artificially elevated sharpness values due to image noise and artifacts, $NSRT_2$ is proposed as an adaptive measure.

The contributions of this paper are: (1) Evaluation of three compression methods, JPEG, JPEG 2000, and Vidware Vision^{TM} by different full- and no-reference objective image quality measures; (2) The design of $NSRT_2$ as an adaptive measure, and its ability to evaluate the sharpness of decompressed images for machine recognition applications.

The remainder of this paper is organized as follow. Section 2 describes our proposed adaptive sharpness measure, $NSRT_2$. Experimental results are demonstrated in section 3, and Section 4 concludes the paper.

2 Sharpness Quality Measure

Sharpness measures are traditionally used to quantify out-of-focus blur. Nevertheless, their extension to evaluate the sharpness of compressed images is non-trivial. To avoid artificially elevated sharpness values due to image noise and artifacts, adaptive sharpness measures assign different weights to pixel gradients according to their local activities. For pixels in smooth areas, small weights are used. For pixels adjacent to strong edges, large weights are allocated. Adaptive sharpness measures are comprised of two determinant factors: the definition of local activities and the selection of weight functions. Figure 1 depicts the flow chart to compute adaptive sharpness measures.

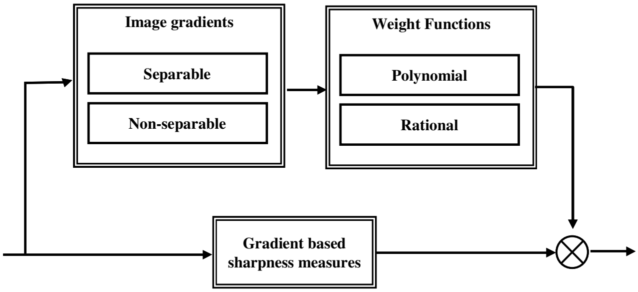


Fig. 1. Flow chart to compute adaptive sharpness measures

Based on how local activities are described, adaptive sharpness measures can be divided into two groups: separable and non-separable. Separable measures only focus on horizontal and vertical edges. A horizontal image gradient $g_x(x, y)$ and a vertical image gradient $g_y(x, y)$ are computed independently. For instance,

$$\begin{cases} g_x(x,y) = f(x+1,y) - f(x-1,y) \\ g_y(x,y) = f(x,y+1) - f(x,y-1) \end{cases} \tag{1}$$

In contrast, non-separable measures include the contributions from diagonal edges. An example of this type of image gradients $g(x, y)$ is given by:

$$g(x,y) = f(x-1,y) + f(x+1,y) - f(x,y-1) - f(x,y+1)\,. \tag{2}$$

Different forms of weights can be used, among which polynomial and rational functions are two popular choices. The polynomial, to be more specific cubic, and rational functions are also exploited in adaptive unsharp masking [14], [15]. The polynomial weights suppress small variation mostly introduced by image noise and have been proved efficient in evaluating the sharpness of high magnification images [16]. The rational weights emphasize a particular range of image gradients. Taking the non-separable image gradient $g(x, y)$ for example, the polynomial weights are:

$$\omega(x,y) = g(x,y)^{p_\omega} \tag{3}$$

where p_ω is a power index determining the degree of noise suppression. The rational weights can be written as

$$\omega(x,y) = \frac{(2k_0 + k_1)g(x,y)}{g^2(x,y) + k_1 g(x,y) + k_0^2} \tag{4}$$

where k_0 and k_1 are coefficients associated with the peak position L_0 and width ΔL of the corresponding function, respectively, and comply with the following equations

$$\begin{aligned} k_0 &= L_0 \\ k_1^2 + 8k_0k_1 + 12k_0^2 - \Delta L^2 &= 0 \end{aligned} \tag{5}$$

Figure 2 illustrates the comparison of different forms of weight functions.

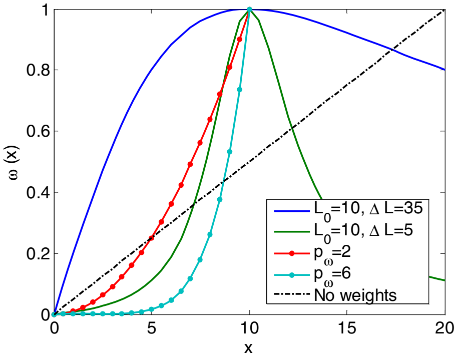


Fig. 2. Illustration of weight functions

The newly designed weights are then applied to gradient based sharpness measures to construct adaptive sharpness measures. For the Tenengrad measure for instance, the resulting separable measure is given by:

$$S = \sum_M \sum_N \left[\omega_x(x,y)g_x^2(x,y) + \omega_y(x,y)g_y^2(x,y)\right] \tag{6}$$

where $\omega_x(x,y)\,/\,\omega_y(x,y)$ denotes the weights obtained from the horizontal/vertical gradients $g_x(x,y)/g_y(x,y)$. For non-separable methods, the corresponding adaptive Tenengrad is formulated as

$$S = \sum_M \sum_N \omega(x,y)\left[g_x^2(x,y) + g_y^2(x,y)\right] \tag{7}$$

To account for blocking artifacts, we follow the method Wang *et al.* proposed in [12]. The above measures are divided into two groups: pixels within a block and pixels at block borders. Accordingly, we compute two values

$$S = \sum_{\substack{x=0,\\ \mathrm{mod}(x,8)\neq 0}}^{M-1} \sum_{\substack{y=0\\ \mathrm{mod}(x,8)\neq 0}}^{N-1} \omega(x,y)\left[g_x^2(x,y)+g_y^2(x,y)\right]$$

and (8)

$$B = \sum_{x=0,}^{\left\lfloor \frac{M}{8} \right\rfloor -1} \sum_{y=0}^{\left\lfloor \frac{N}{8} \right\rfloor -1} \omega(8x,8y)\left[g_x^2(8x,8y)+g_y^2(8x,8y)\right]$$

The final image quality measure combines these two values:

$$Q = S^p B^{-1} \tag{9}$$

The adjustable variables in the rational weight based measures include the peak position L_0, response width ΔL, and weight power p. These parameters can be selected according to the visual perception of the images. In our implementation, pixel gradients $g_x(x,y)/g_y(x,y)/g(x,y)$ are scaled by their mean before computing the weights. The chosen measure NSRT$_2$ then has a peak position k_0 of 10 and a weight power p of 2. The coefficient k_1 is forced to be zero, resulting in a response width of $\Delta L = 2\sqrt{3}L_0 = 35$. In practice, different forms of image gradients are employed for deriving weights, as in equations (3) and (4), and computing sharpness, as in equations (6) and (7), for improved robustness to image noise and artifacts. For weight computation the image gradients used are listed in equations (1) and (2), while for sharpness computation the image gradients based on the Sobel filters are recommended.

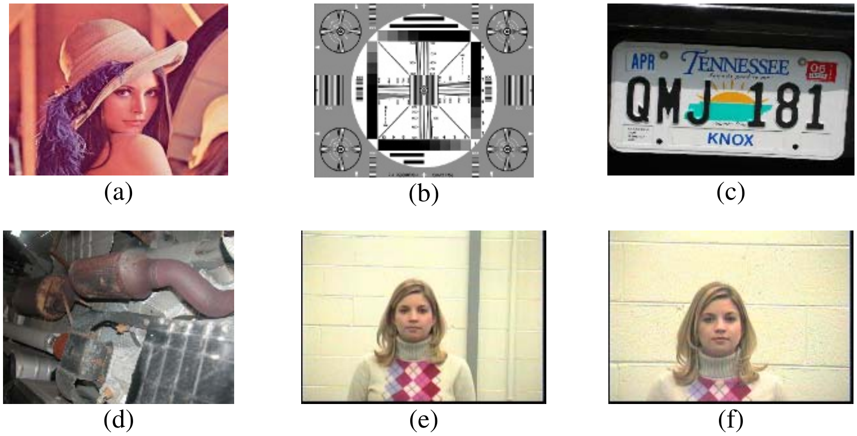
3 Experimental Results

We compare the performance of JPEG, JPEG 2000, and Vidware Vision™ based on a data set composed of compressed images at various compression ratios. The 48 raw color images (24 bits/pixel RGB) in the data set include 12 selected images with different resolutions and 36 long range face images (resolution: 720×480). The 12 selected images, as shown in part in Fig. 3 (a)-(d), can be further divided into 4 categories. The first category includes some well-known test images such as the Baboon, Lena, and Peppers. The second category is comprised of standard test patterns, the IEEE resolution chart (sinepatterns.com), Television Color Test Pattern (high-techproductions.com), and Color Test Template (eecs.berkeley.edu). The third category focuses on images used in face and license plate recognition, two exemplary applications of machine recognition. High resolution images are covered in the last category. The second group, as shown in part in Fig. 3 (e)-(f), is a collection of face images of 6 subjects. A total of 6 images per subject are obtained at various distances

(9.5, 10.4, 11.9, 13.4, 14.6, and 15.9 meters) and with different camera's focal length. The camera's focal length is adjusted so that the subject's face presents similar sizes in all 6 images. Each selected image is compressed by JPEG, JPEG 2000, and Vidware Vision^{TM} with different compression ratios varying from 0.2% to 30%. This produces a total of 342 compressed images for each compression method. In addition, the compression ratio used in this paper indicates:

$$C = \frac{R}{O} \tag{10}$$

Where C, R, and O represent compression ratio, file size of the compressed image, and file size of the original image respectively.



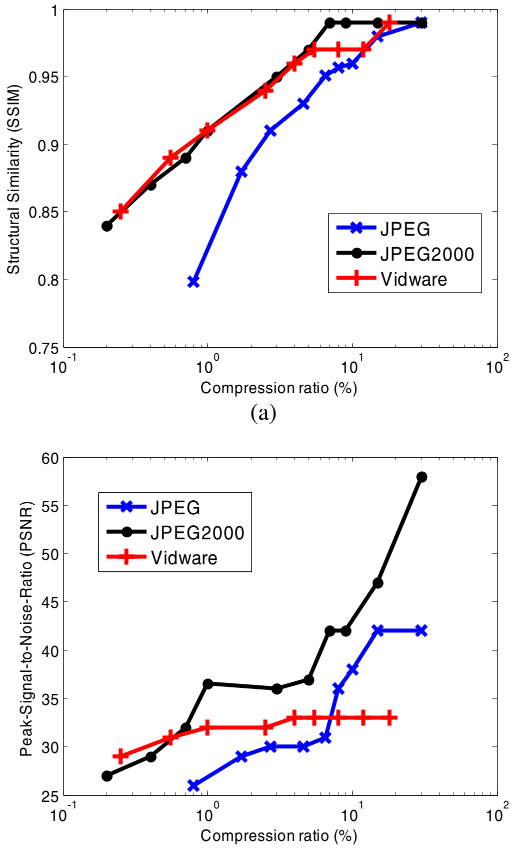
(a) (b) (c)

(d) (e) (f)

Fig. 3. Illustration of test images, (a) Lena 512×512; (b) IEEE resolution chart 1440×1086; (c) License plate 1024×768; (d) Under vehicle view 2560×1920; (e) 9.5 meters face image; (f) 15.9 meters face image

3.1 Full-Reference Quality Measures

Figure 4(a) illustrates the computed SSIM measure. We could see that JPEG 2000 and Vidware Vision^{TM} outperform JPEG for all tested compression ratios. For compression ratios in the range of 7%~10%, JPEG 2000 presents a slightly better performance compared with Vidware Vision^{TM}. However, this performance difference diminishes for other tested compression ratios. Figure 4(b) illustrates the computed PSNR measure. As expected, JPEG 2000 outperforms JPEG for all tested compression ratios. However, their PSNR values increase quickly with respect to decreased compression ratios, indicating a rapidly degraded image quality. In contrast, the PSNR value of Vidware Vision^{TM} increases at a considerably slower speed. As a result, Vidware Vision^{TM} eventually outperforms both JPEG and JPEG 2000 as the compression ratio goes below 0.7%.



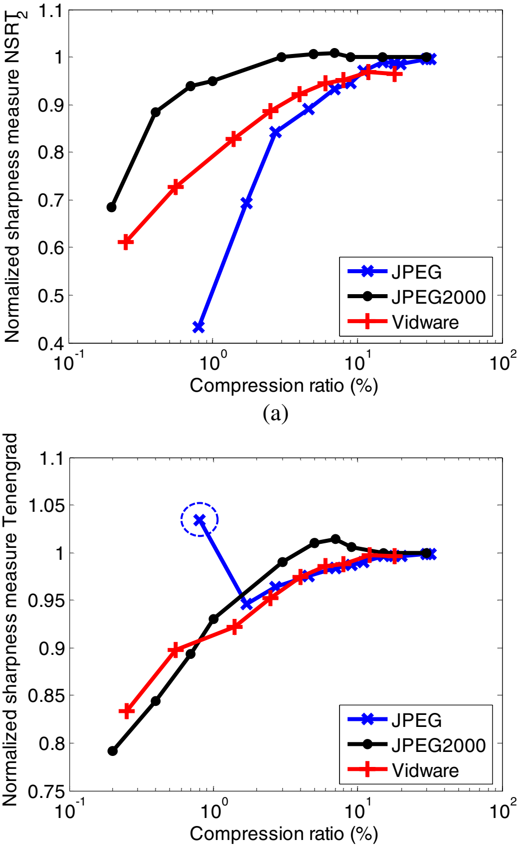
(a)

(b)

Fig. 4. (a)SSIM and (b) PSNR comparison among JPEG, JPEG 2000, and Vidware Vision™

3.2 No-Reference Quality Measures

For fair comparisons, the computed sharpness values are first normalized with respect to the corresponding values of the uncompressed images. Figure 5 shows the performance comparison among JPEG, JPEG 2000, and Vidware Vision™ based on the proposed measure, NSRT_2 and the Tenengrad measure. In Fig. 5(a), we could see similar output based on full-reference measures. JPEG 2000 produces the best performance. For most of the tested compression ratios, the performance of Vidware Vision™ falls in between those of JPEG and JPEG 2000. Exceptions are observed when the compression ratio is larger than approximately 10%, where JPEG outperforms Vidware Vision™ and yields a performance comparable to that of JPEG 2000.



(a)

(b)

Fig. 5. Comparisons among JPEG, JPEG 2000, and Vidware Vision^{TM} based on image sharpness evaluated by (a) NSRT$_2$ and (b) Tenengrad

The conventional Tenengrad measure shown in Fig. 5(b) is also implemented and serves as comparison reference to validate the use of the newly developed NSRT$_{2.}$ The Tenengrad measure is tuned to the local activities of image gradients only and disregards either the sources or the visual effects of these local activities. The artifacts from image compression are also regarded as useful local details, resulting in similar sharpness values for all decompressed images with the same compression ratio. As a consequence, simple Tenengrad measure is insufficient and not applicable. Furthermore, at a compression ratio of 0.8%, as highlighted by a dashed circle in Fig. 5(b), the Tenengrad measure fails because of the overwhelming blocking artifacts. In comparison, the proposed NSRT$_2$ is able to distinguish artifacts from desired local details and thus is qualified to evaluate the performances of tested compression methods. In general, the performance of Vidware Vision^{TM} is less sensitive and decreases gradually with respect to the decreased compression ratio. In comparison, the performance of JPEG deteriorates significantly for compression ratios smaller than 2% and similar behavior occurs to JPEG 2000 for compression ratios smaller than 0.8%.

3.3 Visual Perception

Figure 6 illustrates visual perception of JPEG, JPEG 2000, Vidware Vision^{TM} with 0.7% compression ratio. It is obvious that JPEG has inferior performance, but it is not easy to judge the performance of JPEG 2000 and Vidware Vision^{TM}.

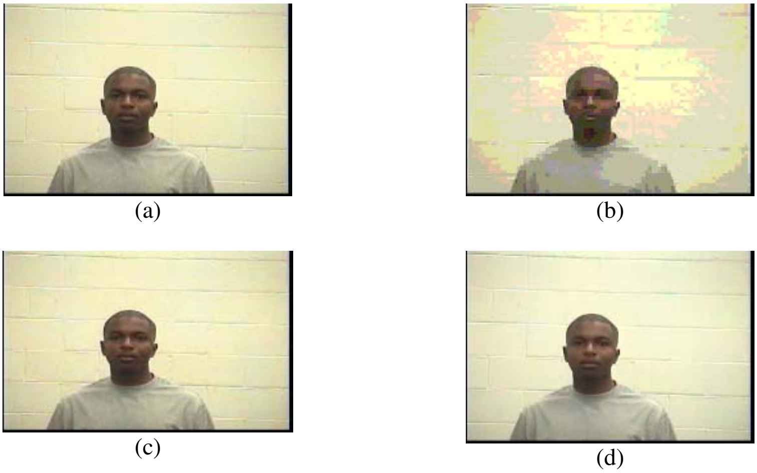

(a)
(b)
(c)
(d)

Fig. 6. Illustration of visual perception with 0.7% compression ratio. (a) original 9.5 meters face image; (b) JPEG; (c)JPEG 2000; (d) Vidware Vision^{TM}.

4 Conclusion

In this paper, we evaluated three different compression methods, JPEG, JPEG 2000, and Vidware Version^{TM} by existing metrics including PSNR, SSIM, and Tenengrad. We also proposed a new image sharpness measure, NSRT$_2$, for evaluating the sharpness of decompressed images. According to our experimental results, we had the following observations: (1) In general, for lower compression ratios (<0.7%), Vidware Vision^{TM} outperforms JPEG 2000. (2) There exists an obvious turning point in compression ratio, beyond which the performances of JPEG and JPEG 2000 begin to degrade rapidly. In contrast, the performance of Vidware Vision^{TM} is less sensitive to the compression ratio and the performance decrease is almost uniformly distributed in the tested compression ratio range. (3) Compared to conventional sharpness measures, our proposed image sharpness measure is robust to image artifacts introduced by image compression and produces a reliable evaluation of the sharpness of decompressed images.

Acknowledgments. This work is supported by the University Research Program in Robotics under grant DOE-DE-FG52-2004NA25589 and by the DOD/RDECOM/NAC/ARC Program under grant W56HZV-04-2-0001. The authors would like to also thank Vidware Incorporated and Databolts Incorporated for kindly providing the compression software used in this study.

References

1. JPEG 2000: Verification Model 4.0, ISO/IEC JTC 1/SC 29/WG 1, Charilaos Christopoulos (Ericsson, Sweden), Editor, April 22 (1999).
2. JPEG-LS: ISO/IEC 14495-1:2000 Information Technology-Lossless and near-lossless compression continuous-tone still images.
3. Vidware Vision^{TM} website, http://www.vidware.com/.
4. Databolts^{TM} website, http://www.databolts.com/.
5. Zhu, L., Zhang, A., Rao, A., and Srihari, R.: Keyblock: An Approach for Content-Based Image Retrieval. ACM Press New York (2000).
6. Zunino, R. and Rovetta, S.: Vector Quantization for License-Plate Location and Image Coding", IEEE Trans. on Industrial Electronics, Vol. 47, No. 1, pp. 159-167, Feb. (2000).
7. Dang, P. P. and Chau, P. M.: Image Encryption for Secure Internet Multimedia Applications. IEEE Transactions on Consumer Electronics, Vol. 46, No. 3, pp. 395-403, Aug. (2000).
8. Lambrecht, C. J. B. and Verscheure, O.: Perceptual Quality Measure Using a Spatial-Temporal Model of Human Visual System. Digital Video Compression Algorithms and Technologies, Proc. SPIE, Vol. 2668, San Jose, (1996).
9. Eskicioglu, A. M. and Fisher, P. S.: Image Quality Measures and Their Performance. IEEE Transactions on Communications, Vol. 43, No. 12, Dec. (1995).
10. Girod, B.: What's Wrong with Mean-Squared Error in Digital Images and Human Vision. MA: MIT Press, pp. 207-220, (1993).
11. Christopoulos, C., Skodras, A., and Ebrahimi, T.: The JPEG 2000 still image coding system: An Overview. IEEE Trans. on Consumer Electronics, Vol. 46, No. 4, pp. 1103-1127, Nov. (2000).
12. Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P.: Image Quality Assessment: from Error Visibility to Structural Similarity. IEEE Trans. on Image Processing, Vol. 13, No. 4, pp. 600-612, April (2004).
13. Santos, A., de Solorzano, C. O., Vaquero. J. J., Pena, J. M., Malpica, N., and del Pozo, F.: Evaluation of Autofocus Functions in Molecular Cytogenetic Analysis. Journal of Microscopy, vol. 188, pp. 264-272, Dec. (1997).
14. Ramponi, G.: A Cubic Unsharp Masking Technique for Contrast Enhancement.: Signal Processing, vol. 67, no. 2, pp. 211-222, Jun. (1998).
15. Ramponi, G. and Polesel, A.: Rational Unsharp Masking Technique. Journal of Electronic Imaging, vol. 7, no. 2, pp. 333-338, April (1998).
16. Yao, Y., Abidi. B. R., and Abidi, M. A.: Digital Imaging with Extreme Zoom: System Design and Image Restoration. IEEE Conf. on Computer Vision Systems, New York, Jan. (2006).

Efficient H.264 Encoding Based on Skip Mode Early Termination

Tien-Ying Kuo and Hsin-Ju Lu

Dept. of Electrical Engineering, National Taipei University of Technology, Taiwan, R.O.C.
tykuo@ee.ntut.edu.tw, hjlu@image.ee.ntut.edu.tw

Abstract. In order to increase the coding efficiency, the H.264 video coding standard introduces the variable block size motion compensation for a better inter frame coding, which adopts seven modes of different block sizes as well as a special mode, the skip mode. Due to the complex nature of the variable block size motion estimation, finding the best inter coding for H.264 would take up a significant complexity. Thus, in this paper, we propose an efficient skip mode detection out of all inter modes to speed up the encoding process. The proposed approach is based on the result of a simple test on 8x8 motion search. The experiment results show that our proposed algorithm is simple yet effective.

Keywords: H.264, variable block size motion compensation, skip mode, early termination.

1 Introduction

The H.264 standard is the latest video codec developed by the Joint Video Team (JVT) [1], which introduces several new coding tools to improve on the rate-distortion performance to the past coding standards. For example, the variable block size motion compensation, the sub-pixel motion estimation, and the multiple reference frame motion compensation are the tools brought in for the inter coding efficiency [2]. However, these new strategies dramatically increase the computational complexity of the encoder in comparison with previous standards. Thus, how to reduce the complexity of encoding process while keeping good coding performance is an important issue on H.264.

The technique of computational complexity reduction in variable block size motion estimation can be defined as two categories. The first category is to apply various efficient search patterns instead of checking every position inside the search range as in the method of full search (FS). These methods [3]-[5] to reduce the search points utilizes search patterns including diamond search (DS) [6], hexagon search [7], or kite-cross-diamond search (KCDS) [8] to improve the encoding speed. Note that, these fast motion estimation algorithms are not fast enough for variable block size motion estimation since seven inter modes are still needed to be examined one by one exhaustively.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 761 – 770, 2006.

The second category is based on the block mode decision, which is to drop impossible modes from motion estimation and thus to achieve complexity reduction. Some works considering block mode decision besides the fast search to speed up the variable block size motion estimation have been proposed [9]-[12]. However, all of the approaches focused on the selection of seven block modes, and did not address the skip mode detection. This is, they have to always perform motion search on at least some of the seven block modes even the skip mode is the best encoding choice. For instance of Lee's [9] method, at least four and up to seven modes still should examine full motion estimation and the large amount computational cost of the motion estimation is still required, and it may waste the complexity when the final mode is selected as skip mode.

In order to save more encoding time, several methods [10][11][12] proposed to check the skip mode for early termination prior to encoding the other modes. In Yin's [10] and Zhang's [11] works, they performed skip mode encoding at first, if the R-D cost of the skip mode is less than a preset threshold, then skip mode is chosen for the macroblock and not turn to the seven mode test. However, a fixed preset threshold is required and difficult to set to cope with different types of video, such as with different motion activity or resolution video. Another skip mode detection method is proposed by Lee [12], Lee first performed 16x16 motion search. If the outcome shows the R-D cost of skip mode is zero, then the skip mode is chosen. This method checked the skip mode in a very strict way such that only fewer cases can fall into the skip mode. Thus, its coding results are inefficient.

The objective of this paper is to investigate an efficient inter-coding based on the skip mode early termination. Unlike the existing methods in literature, our method neither requires any preset threshold nor checks in strictly way to sacrifice the R-D performance. The proposed method can be combined with the existing methods as mentioned in the previous two categories. This paper is organized as follows. The background is introduced in Section 2. The main algorithm will be described in Section 3. Overall experiment results are shown in Section 4. Section 5 concludes the paper.

2 Background – H.264 Inter Modes

As shown in Fig. 1, in H.264, each macroblock in inter mode prediction can be divided into block partition of sizes 16x16, 16x8, 8x16 or 8x8 pixels (i.e., modes 1-4), and the 8x8 block can be further partitioned into sizes of 8x8, 8x4, 4x8 or 4x4 pixels (i.e., modes 4-7), called sub-macroblock partition. So there are as many as 259 kinds of block combination for a macroblock to perform motion search. In JM [13] software implementation, the mode decision is made by comparing the rate-distortion cost (R-D cost) of each partition. The R-D cost function of mode decision is evaluated as,

$$J(\mathbf{m},\lambda) = SAD(s,c(\mathbf{m})) + \lambda \bullet R(\mathbf{m}-\mathbf{p}) . \qquad (1)$$

where $\mathbf{m} = (m_x, m_y)$ denotes the motion vector; $\mathbf{p} = (p_x, p_y)$ denotes the predicted motion vector from neighbors'; λ is the Lagrange multiplier; $R(\mathbf{m}-\mathbf{p})$ represents the rate function of motion information and is computed by table-lookup; SAD represents the sum of absolute difference, and is used as distortion measure,

$$SAD(s,c(\mathbf{m})) = \sum_{y=1}^{M}\sum_{x=1}^{N} | s[x,y] - c[x-m_x, y-m_y] | \quad M,N \in \{16,8,4\} \tag{2}$$

where s denotes the original video signal and c is the coded video signal.

Besides supporting this seven block modes in inter-coding, H.264 can encode the target macroblock as a whole into the skip mode, which signals it is identical to a 16x16 block located in the immediately previous reference frame offset by the predicted motion vector $\mathbf{p}$ as obtained by a median operation on the vectors of the neighboring blocks. Note that, the skip mode is just a signal and no further data, e.g., motion vector and residue (bits to transmitting the quantized prediction error), is presented for the macroblock in the bitstream. Thus, for the macroblock encoded by skip mode, its R-D cost function, as indicated in Eq. (1), becomes a simpler form of $SAD(s,c(\mathbf{p}))$ since no motion vector is transmitted, i.e., $R(\mathbf{m}-\mathbf{p})=0$, and the default motion vector for the skip mode is set as the predicted motion vector, i.e., $\mathbf{m}=\mathbf{p}$. Note that, the skip mode consumes much less complexity than that of seven modes since only the SAD calculation has to be done at the single check point offset by $\mathbf{p}$.

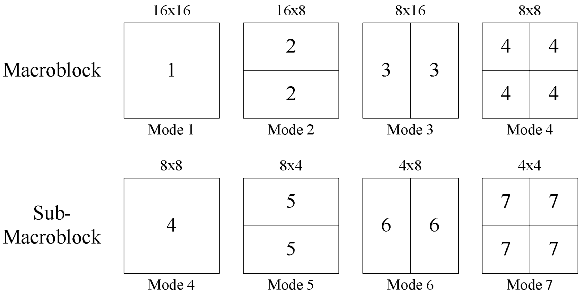


Fig. 1. Seven inter modes for the motion compensation are supported in H.264

3 Proposed Method

3.1 The Main Scheme

The flowchart of the proposed method of skip mode detection (SMD) is illustrated in Fig. 2. In Fig. 2(a), it indicates the global framework of inter-coding adopted in JM, where all seven modes as well as the skip mode are processed and evaluated the cost of Eq. (1) in parallel as one-stage structure. Based on this framework, even some mode selection are performed on the seven block modes [9]-[12], the complexity of motion search on those selected modes may waste when skip mode is chosen. Thus, as shown in Fig. 2(b), the works [10][13] adopted two-stage structure by pulling the skip mode detection on top of the seven mode test. As mentioned previously, a preset threshold is required then.

To solve the preset threshold problem, as shown in Fig. 2(c) , our SMD proposed a three-stage structure by further pulling up the mode 4 (8x8 block) motion search prior to the skip mode detection, for serving as a reference for skip mode decision. That is, at the first stage, our target 16x16 macroblock is disjointed into four blocks of size 8x8 (i.e., mode 4) to perform motion search on the immediate previous frame, to obtain the four motion vectors $V_4^{t-1} = \{V_{4,i}^{t-1} \mid i = 0,1,2,3\}$ and their corresponding minimal R-D costs $\{J(V_{4,i}^{t-1}) \mid i = 0,1,2,3\}$ by Eq. (1), where *i* denotes the block index of mode 4.

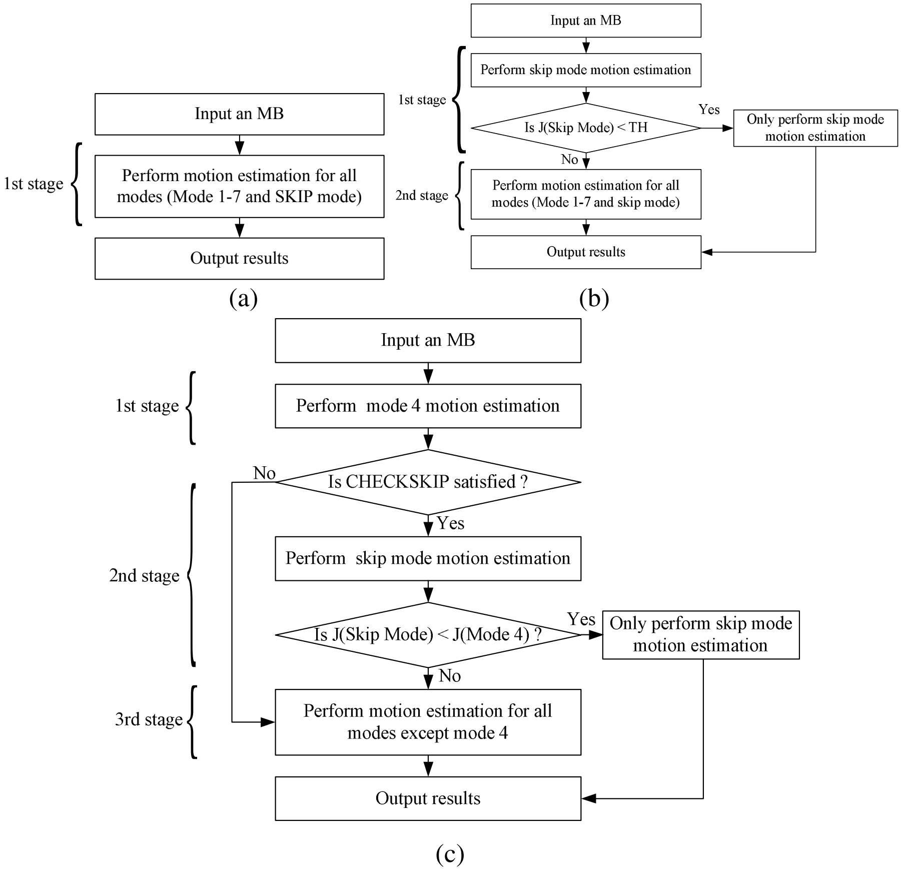

Input an MB
1st stage
Perform motion estimation for all modes (Mode 1-7 and SKIP mode)
Output results
(a)
Input an MB
1st stage
Perform skip mode motion estimation
Is J(Skip Mode) < TH
Yes
Only perform skip mode motion estimation
No
2nd stage
Perform motion estimation for all modes (Mode 1-7 and skip mode)
Output results
(b)
Input an MB
1st stage
Perform mode 4 motion estimation
No
Is CHECKSKIP satisfied ?
Yes
2nd stage
Perform skip mode motion estimation
Is J(Skip Mode) < J(Mode 4) ?
Yes
Only perform skip mode motion estimation
No
3rd stage
Perform motion estimation for all modes except mode 4
Output results
(c)

Fig. 2. The flowchart of the proposed method skip mode decision

Note that, Lee's approach [16] has similar structure as ours, but they performed the mode 1 (16x16 block) motion search instead, and turned to the skip mode only when the mode 1 motion vector is equal to **p** and $J(\mathbf{p}, \lambda) = 0$. The R-D cost efficiency of

Lee's approach would be poor because it is often to have the case that $J(\mathbf{p},\lambda)$ is not equal to zero but is still less than all costs of the seven modes. On the contrary, in our proposed SMD, the skip mode is determined under some condition tests based on motion field of mode 4 as well as its R-D cost comparison with skip mode. Note that, if the skip mode is decided, a huge complexity could be saved since no further test on the rest modes is required.

Next, we would like to demonstrate why the early termination by skip mode detection is necessary, and why mode 4 is selected for motion search at the beginning. Fig. 3 indicates the statistical analysis of mode distributions resulting from the full search on several video sequences with various QPs on average. From this figure, we observe that, the majority of the mode distribution falls into skip mode and occupies even up to 50% to 70% of all distributions in low or spark motion sequences. This proves that the second stage of skip mode detection is necessary, because the complexity saving can be achieved as the high chance of skip mode early termination. Moreover, it shows that mode 4 (i.e., size 8x8) is more often decided as the best mode when the skip mode is not taken into account. Thus, it is reasonable for us to perform motion search on mode 4 at first stage, and design our algorithm based on that.

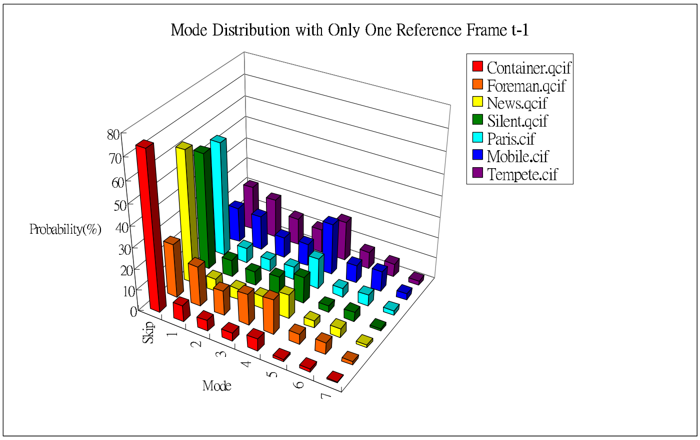


Fig. 3. Mode distribution with only one reference frame

3.2 Skip Mode Detection

Generally speaking, skip mode usually occurs when the contents of video sequences are at a standstill or with very slow motion, such as the static background. Thus, we define the condition, CHECKSKIP, to detect the skip mode at the second stage by checking the length of motion vectors V_4^{t-1} given by the first stage. The skip mode is activated when CHECKSKIP happens, where at least three of the four motion vectors in V_4^{t-1} are zero and the R-D cost of skip mode is less or equal to that of mode 4, i.e.,

$J(Skip\ Mode) < J(Mode\ 4)$ and $J(Mode\ 4) = \sum_{i=0}^{i=3} J(V_{4,i}^{t-1})$ with $J(\cdot)$ as defined in Eq. (1). Note that, since skip mode decision made by merely utilizing the motion activity of V_4^{t-1} may not be very reliable. We improve the detection precision by adding the R-D cost comparisons between skip mode and mode 4, where the R-D cost of skip mode is easy to get while that of mode 4 should be available already during the mode 4 motion search.

Next, we evaluate whether CHECKSKIP is a good metrics or not. In Table 1, P(skip) represents the probability of the actual skip mode given by full search, which has been plotted as the first bar in Fig. 3. In this table, P(skip|CHECKSKIP) and P(CHECKSKIP|skip) are also shown. We would expect both probabilities are close to 100% as the likeness increases between skip mode and CHECKSKIP. Let us first look at the value of P(CHECKSKIP|skip), we observe it is about 73%-84% for the sequences with the low or spark motion. That means, for those sequences, CHECKSKIP does the job to catch around 80% of the skip mode, that is, the hit rate is high. Though, for some video sequence, such as Foreman, Mobile and Tempete, P(CHECKSKIP|skip) is low as only 2%-33%, it does not affect the performance too much since those sequence has also low skip mode usage as indicated by P(skip). And since those sequences all are with global camera motion or half-pel motion, thus encoding them with other modes instead of skip mode is still reasonable because the PSNR difference would be small as shown later in next section. Next we look at the probability P(skip|CHECKSKIP), when the input macroblock matches the condition of CHECKSKIP, the probability of skip mode occurs about 83.44% on average. This means the CHECKSKIP mode is efficient and the case of false alarm to the skip mode is quite few. Thus, we conclude that using CHECKSKIP to identify the skip mode is satisfied.

Table 1. Each probability in condition CHECKSKIP

Sequence	P(skip)	P(skip|CHECKSKIP)	P(CHECKSKIP|skip)
Container	74.36	89.09	84.30
Foreman	25.56	80.97	29.95
News	62.48	93.32	84.50
Silent	55.37	92.67	73.97
Paris	54.53	92.43	80.04
Mobile	17.08	56.28	2.27
Tempete	21.34	79.30	33.12
Average	**44.39**	**83.44**	**55.45**

4 Experiment Results

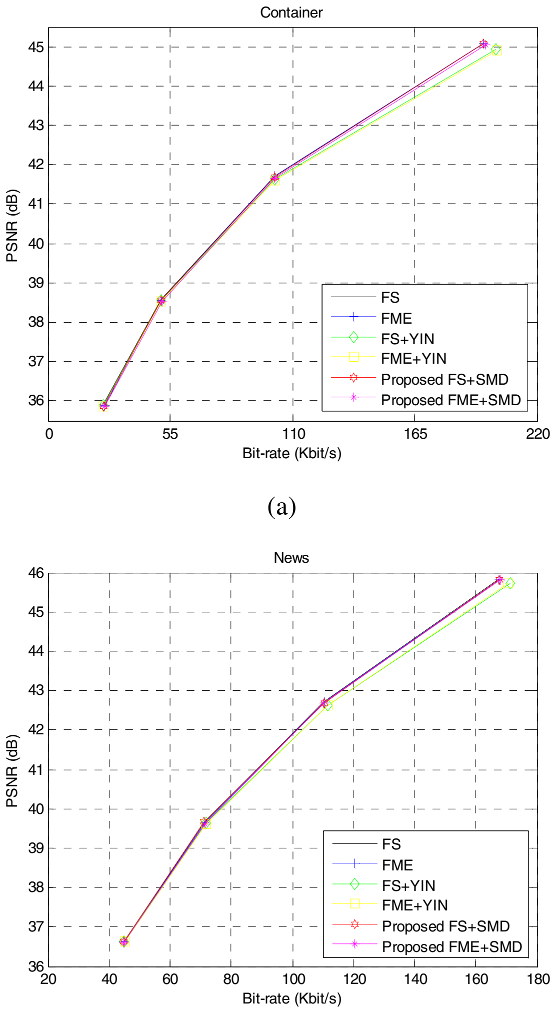
Our experimental environment is based on H.264 reference encoder of Joint Mode (JM) 9.2 [14]. Experimental results are tested with the conditions indicated in Table 2, which strictly follows the simulation suggestion of JVT [15], specifying and covering seven video sequences of different motion activity, resolutions (QCIF and CIF), frame rates/frame skipping (10-30 f.p.s.) and quantization parameters. We encode the first frame of video as an I-frame and the rest as P-frames with all seven block modes activated. The R-D optimization is turned on and the multiple reference frames are turned off.

Table 2. Encoder parameters used in experiments

Sequence/ Resolution/ Total frames/ Frame rate (Hz)	Container	qcif	300 frames	10 Hz
	News	qcif	300 frames	10 Hz
	Foreman	qcif	300 frames	10 Hz
	Silent	qcif	300 frames	15 Hz
	Paris	cif	300 frames	15 Hz
	Mobile	cif	300 frames	30 Hz
	Tempete	cif	260 frames	30 Hz
QPs	16, 20, 24, 28			
Coding options	IntraPeriod		Only first	
	UseHadamard		Enable	
	NumberReferenceFrames		1	
	SearchRange		16	
	RDOptimization		Enable	
	SymboMode		CAVLC	
Codec	JM 9.2 encoder			

Since our proposed SMD can work with any existing motion search algorithm, we adopt two search methods in our SMD test, including the full search (FS) and the hexagonal-based fast motion estimation (FME) [4][5] as implemented in JM 9.2. Note that, both FS and FME exams all modes such that R-D cost for all modes including the skip mode should be completely calculated by Eq. (1), and therefore the huge amount of computational complexity is required. In addition, a literature method YIN [10] involving the skip mode detection is also tested for comparisons. Though the existing algorithms of fast modes 1-7 decision can work with our SMD method, we do not incorporate them here to simply focus on the contribution of our skip mode decision. The R-D curves of each method for two sequences are plotted in Fig. 4, to which other sequences have similar R-D curves but not plotted for limited space. Fig. 4 shows that the R-D performance of our FS+SMD and FME+SMD is as close as FS and FME. Since the curves are very close each other for most cases, we use the BDPSNR (Bjontegaard delta PSNR) and BDBR (Bjontegaard delta bitrate) [16] recommended by JVT to measure the performance difference between methods, which basically calculate the average PSNR and bitrate distance between two R-D curves of two methods, respectively. Table 3 shows the BDPSNR and BDBR of four methods using FS as the comparing basis, because the performance of full search theoretically is the metrics for the upper bound. The negative BDPSNR or positive BDBR indicates the coding loss to FFS and is not preferred.

Table 3 shows that, on average, our proposed SMD algorithm would degrade BDPSNR 0.01dB to FS, and 0.04dB to FME. Such insignificant degradation will not cause noticeable visual difference. As to the bit rate, the average percentage of increase is also as small as 0.37% and 0.90% (i.e., 1.00%-0.10%) to FS and FME, respectively. In contrast to YIN's methods which degrade BDPSNR 0.09dB and 1.11% increasing in BDBR to FS, the results prove our SMD nearly the same coding efficiency as the exhaustive methods. Especially, from Fig. 4, we observe that the performance of YIN becomes significantly worse in the high bit rate case (small QP), while our SMD keep close to the performance of FS and FME for all range of bit rates.



(a)

(b)

Fig. 4. Rate-Distortion curves comparisons among FS, FME, FS+YIN [10], FME+YIN [10] and two proposed methods FS+SMD, FME1+SMD

Table 3. R-D performance of FME, FS+YIN [10], FME+YIN [10] and two proposed methods FS+SMD, FME+SMD (FS is the comparing basis)

Method	FS				FME					
	YIN		Proposed SMD		FME only		YIN		Proposed SMD	
(dB)/(%)	BD PSNR	BD BR	BD PSNR	BD BR	BD PSNR	BD BR	BD PSNR	BD BR	BD PSNR	BD BR
Container	-0.14	1.64	-0.02	0.67	-0.02	0.54	-0.16	2.57	-0.08	1.96
Foreman	-0.15	1.49	-0.02	0.28	0.02	-0.23	-0.16	1.71	-0.06	0.76
News	-0.06	0.76	0.00	0.09	-0.03	0.41	-0.16	1.97	0.00	0.50
Silent	-0.09	0.81	0.00	-0.08	-0.01	0.16	-0.12	1.48	-0.06	0.83
Paris	-0.09	0.92	-0.01	0.21	0.00	0.10	-0.10	1.34	-0.06	1.05
Mobile	-0.01	0.08	0.00	0.00	0.02	-0.34	0.00	0.02	0.00	0.01
Tempete	-0.09	2.06	-0.04	1.41	0.00	0.06	-0.09	2.28	-0.06	1.91
Average	**-0.09**	**1.11**	**-0.01**	**0.37**	**0.00**	**0.10**	**-0.11**	**1.63**	**-0.04**	**1.00**

Next we will discuss the computation complexity of each method in Table 4. Table 4 lists the reduction motion estimation time of each method to FS. As shown in Table 4, our SMD can reduce FS by 27.1% of motion estimation encoding time, which reduces more than that of FS+YIN (22.5%). If our SMD is used with FME, the reduction time to FS even up to 83.1% on average, and is still faster than YIN's method.

Table 4. Percentage of reduction time in FME, FS+YIN [10], FME+YIN [10] and two proposed method FS+SMD, FME+SMD (FS is the comparing basis)

Method Reduction to FS (%)	FS		FME		
	YIN	Proposed SMD	FME only	YIN	Proposed SMD
Container	35.9	37.5	69.9	86.0	87.8
Foreman	9.1	7.4	69.9	78.6	77.4
News	41.2	49.0	65.8	85.1	86.9
Silent	27.5	40.8	65.8	81.1	85.4
Paris	29.6	46.2	67.0	83.1	87.0
Mobile	3.8	4.8	72.3	79.1	78.4
Tempete	10.7	3.8	71.2	80.2	78.7
Average	**22.5**	**27.1**	**68.8**	**81.9**	**83.1**

5 Conclusion

An efficient skip mode decision for H.264 is proposed, by which the early termination of the skip mode can alleviate the huge complexity resulting from the variable block mode partitions. The proposed SMD is to decide the skip mode based on the motion vectors of 8x8 blocks. Compared with the literature methods, the experiments demonstrate that the proposed SMD method consumes lower complexity while gives a better R-D performance at all range of bit rates, especially significantly outperforms at the high bit. Further, no preset threshold required in our SMD method makes it easy to cope with different characteristics of video. Thus, we conclude that the proposed SMD is more effective than the literature methods.

Acknowledgement

This work was supported by the National Science Council (NSC 94-2213-E-027-015) and (NSC 94-2219-E-002-012).

References

1. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,: Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), ITU-T | ISO/IEC, 2003.
2. Ostermann, J., Bormans, J., List, P., Marpe, D., Narroschke, M., Pereira, F., Stockhammer, T., Wedi, T.: Video coding with H.264/AVC: tools, performance, and complexity, IEEE Circuits Syst. Magazine, Vol. 4, no. 1, First Quarter 2004, pp. 7-28.
3. Zhu, C., Lin, X., Chau, L.P., Po, L.M.: Enhanced hexagonal search for fast block motion estimation, IEEE Trans. Circuits Syst. Video Technol., Vol. 14, no. 10, Oct. 2004, pp. 1210-1214.
4. Chen, Z., Zhou, P., He, Y.: Fast Integer Pel and Fractional Pel Motion Estimation for JVT, ITU-T Q6/SG16, Doc#JVT-F017, 5-12 Dec. 2002.
5. Chen, Z., Zhou, P., He, Y.:Fast Motion Estimation for JVT, ITU-T Q6/SG16, Doc#JVT-G016, 7-14 March 2003.
6. Zhu, S., Ma, K.K.: A new diamond search algorithm for fast block-matching motion estimation, IEEE Trans. Image Processing, Vol. 9, no. 2, Feb. 2000, pp. 287-290.
7. Zhu, C., Lin, X., Chau, L.P., Lim, K.P., Ang, H.A., Ong, C.Y.: A novel hexagon-based search algorithm for fast block motion estimation, in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Vol. 3, May 2001, pp. 1593-1596.
8. Lam, C.W., Po, L.M., Cheung, C.H.: A novel kite-cross-diamond search algorithm for fast block matching motion estimation, in Proc. IEEE Int. Symposium Circuits and Systems, Vol. 3, May 2004, pp. 729-732.
9. Lee, J., Jeon, B.: Fast Mode Decision for H.264 with Variable Motion Block Sizes, in Proc. IEEE Int. Symposium Computer and Information Sciences, Vol. 2869, Nov. 2003, pp.723-730.
10. Yin, P., Tourapis, H.-Y.C., Tourapis, A.M., Boyce, J.: Fast mode decision and motion estimation for JVT/H.264, in Proc. IEEE Int. Conf. Image Processing, Vol.3, Sep. 2003, pp. 853-856.
11. Zhang, D., Shen, Y., Lin, S., Zhang, Y.: Fast Inter Frame Encoding Based on Modes Pre-Decision in H.264, in Proc. IEEE Int. Conf. Multimedia and Expo, Vol. 1, 5-8 July 2005. pp.550-533.
12. Lee, J., Jeon, B.: Pruned Mode Decision based on Variable Block Sizes Motion Compensation for H.264, in Proc. MIPS International Workshop on Multimedia Interactive Protocols and Systems, Vol. 2899, Nov. 2003.
13. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,: Working draft number 2, revision 2 (WD-2), Joint Video Team (JVT), 2002.
14. JM Reference Software 9.2 Available: http://bs.hhi.de/~suegring.html/.
15. Bjontegaar, G.: Recommended Simulation Condition for H.26L, ITU-T Q6/SG16, Doc. #VCEG-L38, 9-12 Jan, 2001.
16. Bjontegaar, G.: Calculation of Average PSNR Differences between RD-curves, ITU-T Q6/SG16, Doc.#VCEG-M33, 2-4 April 2001.

Low-Power H.264 Deblocking Filter Algorithm and Its SoC Implementation

Byung-Joo Kim, Jae-Il Koo, Min-Cheol Hong, and Seongsoo Lee

School of Electronic Engineering, Soongsil University, Seoul 156-743 Korea*
sslee@ssu.ac.kr

Abstract. This paper proposed a low-power H.264 deblocking filter algorithm. In H.264 deblocking filter, filtering can be skipped on some pixels when pixel differences satisfy some specific conditions. Furthermore, whole filtering can be skipped when quantization parameter is less than 16. By exploiting this feature, whole deblocking filter or its some parts can be deactivated during execution, and its power consumption can be significantly reduced up to 20.3%. A low-power H.264 deblocking filter architecture was also proposed. Simple control circuit can totally or partially deactivate deblocking filter, and common hardware performs both horizontal and vertical filtering. The proposed low-power deblocking filter was implemented in silicon chip using 0.35 μm standard cell technology. The gate count is about 20,000 gates. The maximum operation frequency is 108 MHz. The maximum throughput is 30 frame/s with CCIR601 image format.

1 Introduction

H.264 video compression [1] is the newest international standard for video compression. It outperforms previous video compression algorithms in rate-distortion efficiency. It is widely used in many multimedia applications. Deblocking filter is one important tool to achieve higher compression ratio and better subjective quality. H.264 exploits 4×4 block-based integer transform and variable block size motion compensation. It shows a different quantization error distribution with 8×8 block-based discrete cosine transform, and it shows a different blocking artifact with conventional video compressions such as MPEG-2 [2] and MPEG-4 [3]. Consequently, it requires a novel deblocking filter scheme.

In H.264, deblocking filter exists both in encoder and decoder. Since the deblocking filter is more complicated than ordinary filters, it requires quite a large computation. Especially, it occupies considerable amount of computation and power consumption in the decoder [4]. In mobile multimedia broadcasting applications such as digital multimedia broadcasting (DMB), only decoder is used in the mobile multimedia terminal. Therefore, low-power H.264 deblocking filter is essential in the hardware implementation of mobile multimedia terminal. In this case, power consumption is one of the primary concerns, since battery operation time is the key to commercial success [5].

* This work was supported by the Soongsil University Research Fund.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 771–779, 2006.

In this paper, we proposed a low-power H.264 deblocking filter algorithm. It reduces computation and power consumption by totally or partially deactivating deblocking filter whenever some skipping conditions hold. We also proposed a low-power H.264 deblocking filter architecture and implemented it in silicon chip.

2 Low-Power H.264 Deblocking Filter Algorithm

In H.264, blocking artifact occurs in 4×4 block, and it goes worse in 16×16 macroblock boundary. H.264 deblocking filter performs 1-dimensional filtering in horizontal direction first, and then it performs 1-dimensional filtering in vertical direction. Different filtering method is applied whether the pixel locates in the macroblock boundary or it does not.

Figs. 1 and 2 show the H.264 deblocking filter algorithm. First, B_s is determined as shown in Fig. 1, where p_0, p_1, p_2, p_3, q_0, q_1, q_2, and q_3 are original pixels, and P_0, P_1, P_2, P_3, Q_0, Q_1, Q_2, and Q_3 are filtered pixels. Then, filtering is performed based on B_s as shown in Fig. 2, where parameters α and β are given as Fig. 3, and parameter t_{c0} is given as Fig. 4. $B_s = 0$ is *no filtering mode*, and no filtering is performed. $B_s = 1$, 2, or 3 is *standard mode*, and filtering is performed on 0, 2, or 4 pixels based on pixel values and parameters. $B_s = 4$ is *strong mode*, and filtering is performed on 0, 2, or 6 pixels based on pixel values and parameters.

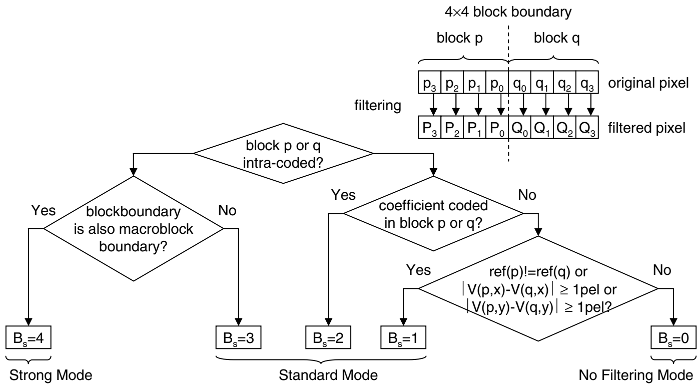
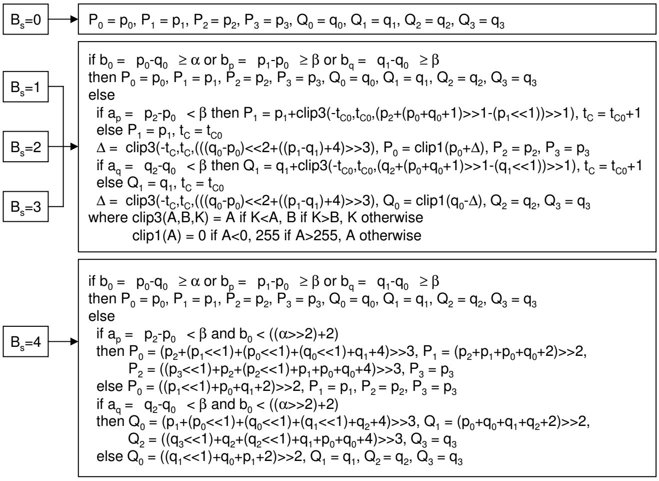


Fig. 1. Determination of B_s

From Figs. 1-4, it is clear that filtering is not applied to some pixels when $t_{c0} = 0$ or $a_p \geq \beta$ or $a_q \geq \beta$. Fig. 5 illustrates the operation of deblocking filter, where O and X mean that the corresponding pixel is filtered or not, respectively. Therefore, computation can be significantly reduced by skipping filtering when condition X in Fig. 5 is satisfied. Furthermore, no computation is required when quantization parameter QP is less than 16, since $\alpha = \beta = 0$. Therefore, the proposed H.264 deblocking filter checks QP first, and it is totally deactivated



```
$P_0 = p_0, P_1 = p_1, P_2 = p_2, P_3 = p_3, Q_0 = q_0, Q_1 = q_1, Q_2 = q_2, Q_3 = q_3$
```

```
if $b_0 = \; p_0\text{-}q_0 \; \geq \alpha$ or $b_p = \; p_1\text{-}p_0 \; \geq \beta$ or $b_q = \; q_1\text{-}q_0 \; \geq \beta$
then $P_0 = p_0, P_1 = p_1, P_2 = p_2, P_3 = p_3, Q_0 = q_0, Q_1 = q_1, Q_2 = q_2, Q_3 = q_3$
else
  if $a_p = \; p_2\text{-}p_0 \; < \beta$ then $P_1 = p_1\text{+clip3}(\text{-}t_{C0},t_{C0},(p_2+(p_0+q_0+1)>>1\text{-}(p_1<<1))>>1), t_C = t_{C0}+1$
  else $P_1 = p_1, t_C = t_{C0}$
  $\Delta = \text{clip3}(\text{-}t_C,t_C,(((q_0\text{-}p_0)<<2+((p_1\text{-}q_1)+4)>>3), P_0 = \text{clip1}(p_0+\Delta), P_2 = p_2, P_3 = p_3$
  if $a_q = \; q_2\text{-}q_0 \; < \beta$ then $Q_1 = q_1\text{+clip3}(\text{-}t_{C0},t_{C0},(q_2+(p_0+q_0+1)>>1\text{-}(q_1<<1))>>1), t_C = t_{C0}+1$
  else $Q_1 = q_1, t_C = t_{C0}$
  $\Delta = \text{clip3}(\text{-}t_C,t_C,(((q_0\text{-}p_0)<<2+((p_1\text{-}q_1)+4)>>3), Q_0 = \text{clip1}(q_0\text{-}\Delta), Q_2 = q_2, Q_3 = q_3$
where clip3(A,B,K) = A if K<A, B if K>B, K otherwise
        clip1(A) = 0 if A<0, 255 if A>255, A otherwise
```

```
if $b_0 = \; p_0\text{-}q_0 \; \geq \alpha$ or $b_p = \; p_1\text{-}p_0 \; \geq \beta$ or $b_q = \; q_1\text{-}q_0 \; \geq \beta$
then $P_0 = p_0, P_1 = p_1, P_2 = p_2, P_3 = p_3, Q_0 = q_0, Q_1 = q_1, Q_2 = q_2, Q_3 = q_3$
else
  if $a_p = \; p_2\text{-}p_0 \; < \beta$ and $b_0 < ((\alpha>>2)+2)$
  then $P_0 = (p_2+(p_1<<1)+(p_0<<1)+(q_0<<1)+q_1+4)>>3, P_1 = (p_2+p_1+p_0+q_0+2)>>2,$
        $P_2 = ((p_3<<1)+p_2+(p_2<<1)+p_1+p_0+q_0+4)>>3, P_3 = p_3$
  else $P_0 = ((p_1<<1)+p_0+q_1+2)>>2, P_1 = p_1, P_2 = p_2, P_3 = p_3$
  if $a_q = \; q_2\text{-}q_0 \; < \beta$ and $b_0 < ((\alpha>>2)+2)$
  then $Q_0 = (p_1+(p_0<<1)+(q_0<<1)+(q_1<<1)+q_2+4)>>3, Q_1 = (p_0+q_0+q_1+q_2+2)>>2,$
        $Q_2 = ((q_3<<1)+q_2+(q_2<<1)+q_1+p_0+q_0+4)>>3, Q_3 = q_3$
  else $Q_0 = ((q_1<<1)+q_0+p_1+2)>>2, Q_1 = q_1, Q_2 = q_2, Q_3 = q_3$
```

Fig. 2. Deblocking filtering

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
α	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2

QP	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
α	5	6	7	8	9	10	12	13	15	17	20	22	25	28	32	36	40	45
β	2	3	3	3	3	4	4	4	6	6	7	7	8	8	9	9	10	10

QP	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	
α	50	56	63	71	80	90	101	113	127	144	162	182	203	226	255	255	
β	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18	

Fig. 3. Parameters α and β

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Bs=1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bs=2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bs=3,4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

QP	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Bs=1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	2	2	2
Bs=2	0	0	0	1	1	1	1	1	1	1	1	1	1	2	2	2	2	3
Bs=3,4	1	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	4	4

QP	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	
Bs=1	2	3	3	3	4	4	4	5	6	6	7	8	9	10	11	13	
Bs=2	3	3	4	4	5	5	6	7	8	8	10	11	12	13	15	17	
Bs=3,4	4	5	6	6	7	8	9	10	11	13	14	16	18	20	23	25	

Fig. 4. Parameter t_{c0}

when $QP < 16$. Then, it checks skipping conditions in Fig. 5 and corresponding hardwares are partially deactivated. The control circuit for Fig. 5 can be implemented by simple combinational logic gates with small hardware overhead.

pixel	$b_0 \geq \alpha$ or $b_p \geq \beta$ or $b_q \geq \beta$	$b_0 < \alpha$ and $b_p < \beta$ and $b_q < \beta$					
		$B_s=0$	$B_s=1,2,3$			$B_s=4$	
			$t_{c0}=0$	$t_{c0} \neq 0$		$a_p < \beta$ and $b_0 < (\alpha >> 2)+2$	$a_p \geq \beta$ or $b_0 \geq (\alpha >> 2)+2$
				$a_p < \beta$	$a_p \geq \beta$		
P_0	X	X	X	O	O	O	O
P_1	X	X	X	O	X	O	X
P_2	X	X	X	X	X	O	X
P_3	X	X	X	X	X	X	X

(a)

pixel	$b_0 \geq \alpha$ or $b_p \geq \beta$ or $b_q \geq \beta$	$b_0 < \alpha$ and $b_p < \beta$ and $b_q < \beta$					
		$B_s=0$	$B_s=1,2,3$			$B_s=4$	
			$t_{c0}=0$	$t_{c0} \neq 0$		$a_q < \beta$ and $b_0 < (\alpha >> 2)+2$	$a_q \geq \beta$ or $b_0 \geq (\alpha >> 2)+2$
				$a_q < \beta$	$a_q \geq \beta$		
Q_0	X	X	X	O	O	O	O
Q_1	X	X	X	O	X	O	X
Q_2	X	X	X	X	X	O	X
Q_3	X	X	X	X	X	X	X

(b)

Fig. 5. Skipping conditions of the proposed low-power H.264 deblocking filter. (a) P_0, P_1, P_2, and P_3. (b) Q_0, Q_1, Q_2, and Q_3.

Fig. 6 shows the computation and power reduction of the proposed low-power H.264 deblocking filter algorithm. Fig. 6 (a), (b), and (c) show the average quantization parameters, the computation time, and the power consumption in "Akiyo" image, respectively. Fig. 6 (d), (e), and (f) show the average quantization parameters, the computation time, and the power conumption in "Container" image, respectively. JM 9.2 baseline profile encoder is used in the simulation. The image format is QCIF (176×144 pixels, 30 frame/s), and 100 frames are tested in both test sequences. Bitrate is varied from 50 kbps to 400 kbps. We designed the conventional and the proposed H.264 deblocking filters for accurate power simulation of real filtering operation. We used Synopsys Prime Power in cycle-by-cycle power simulation, whose procedure is described in [6] in detail.

As shown in Fig. 6, power consumption is reduced up to 24.7% and 20.3% in "Akiyo" and "Container" images, respectively. In Fig. 6 (b) and (e), the computation time and the power consumption increase along with bitrate in the conventional H.264 deblocking filter, while they decrease when bitrate exceeds 200 kbps $\sim$ 250 kbps in the proposed low-power H.264 deblocking filter, respectively. This is due to the following reasons.

When bitrate increases, less coefficients are truncated into zero in the quantization process, because quantization parameter QP decreases in higher bitrate as shown in Fig. 6 (a) and (d). When the number of non-coded coefficients decreases, the number of no filtering mode pixels ($B_s = 0$) decreases. In the conventional H.264 deblocking filters, it skips filtering operations only when the pixel is in the no filtering mode. Therefore, the required computation and power of the conventional H.264 deblocking filters increase along with bitrate as shown in Fig. 6 (b), (c), (e), and (f).

On the contrary, when quantization parameter QP decreases, α and β also decrease as shown in Fig. 3. Therefore, skipping conditions in Fig. 5 occur more

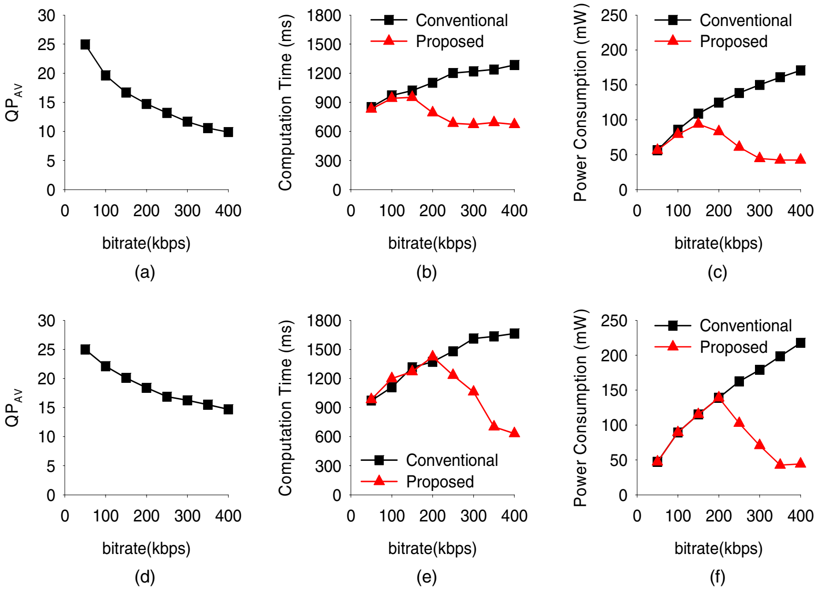


Fig. 6. Computation and power reduction of the proposed low-power H.264 deblocking filter. (a) (b) (c) The average quantization parameters, the computation time, and the power consumption in "Akiyo" image, respectively. (d) (e) (f) The average quantization parameters, the computation time, and the power consumption in "Container" image, respectively.

frequently in the proposed H.264 deblocking filter. Especially, when $QP < 16$, $\alpha = \beta = 0$, and the proposed deblocking filter never performs filtering operations. This significantly reduces the computation and power as shown in Fig. 6 (b), (c), (e), and (f). Note that the average quantization parameter $QP_{AV} < 16$ when bitrate exceeds 200 kbps $\sim$ 250 kbps in the proposed H.264 deblocking filter, and the computation and power start to decreases by skipping filtering operation.

3 Low-Power H.264 Deblocking Filter Hardware

H.264 deblocking filter (DF) performs two-step filtering in macroblock basis, as shown in Fig. 7 (a). When current macroblock C is processed, it requires its left macroblock L in horizontal filtering. In vertical filtering, it requires its upper macroblock U. These macroblocks are motion-compensated pixels from motion compensator (MC), and they are stored in the frame memory (FM). Therefore, MC, FM, and DF are connected via 32-bit AMBA AHB bus, as shown in Fig. 7 (b). The proposed low-power H.264 deblocking filter architecture is illustrated in Fig. 7 (c). It consists of filter, PREG, QREG, and TBUF. Filter processes four 8-pixel filtering simultaneously. PREG and QREG store 4×4 blocks of p and q in Fig. 1, respectively. TBUF is 16×16 pixel buffer with transpose. It stores horizontal filtering results with transpose order, and they are reused in vertical filtering.

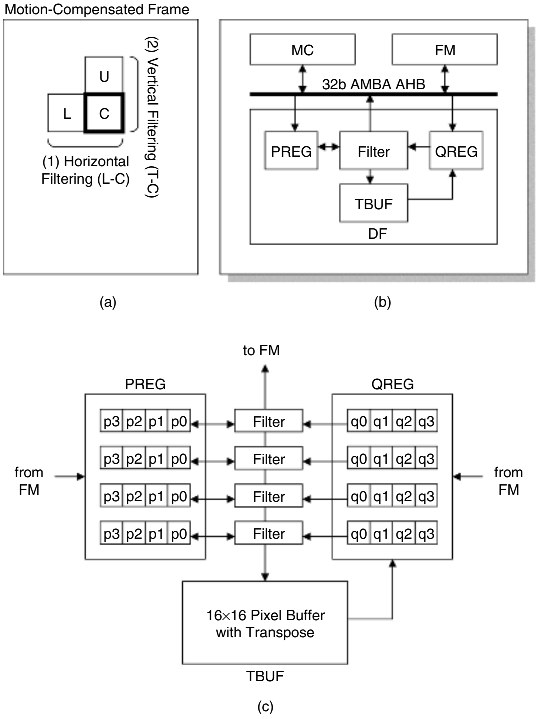


Fig. 7. Architecture of the proposed H.264 deblocking filter. (a) Horizontal and vertical filtering. (b) Deblocking filter, frame memory, and motion compensator. (c) Deblocking filter architecture.

Fig. 8 illustrates horizontal filtering. Current macroblock *C* has four vertical edges *a*, *b*, *c*, and *d* to be processed, as shown in Fig. 8 (a). Pixel data of left macroblock *L* are required to process vertical edge *a*. 4×4 blocks are filtered from block 0 to block 15, i.e. (block -13, block 0), (block 0, block 1), (block 1, block 2), (block 2, block 3), (block -9, block 4), ... (block 14, block 15) are filtered sequentially.

Horizontal filtering is illustrated in detail in Fig. 8 (b). When (block -13, block 0) are processed, block -13 and 0 are read from FM to PREG and QREG, respectively (①). Then, PREG and QREG are filtered (②). Filtered outputs of block -13 are stored in FM (③). At the same time, filtered outputs of block 0 are transfered to PREG for next block filtering (③). At the same time, block 1 is read from FM to QREG for next block filtering (③). When (block 0, block 1) are processed, PREG and QREG are filtered (④). Filtered outputs of block 0 are stored in TBUF for vertical filtering (⑤). At the same time, filtered outputs of block 1 are transfered to PREG for next block filtering (⑤). At the same time, block 2 is read from FM to QREG for next block filtering (⑤). When (block 1, block 2) are processed, PREG and QREG are filtered (⑥). Filtered outputs of block 1 are stored in TBUF for vertical filtering (⑦). At the same time, filtered outputs of block 2 are transfered to PREG for next block filtering (⑦). At the

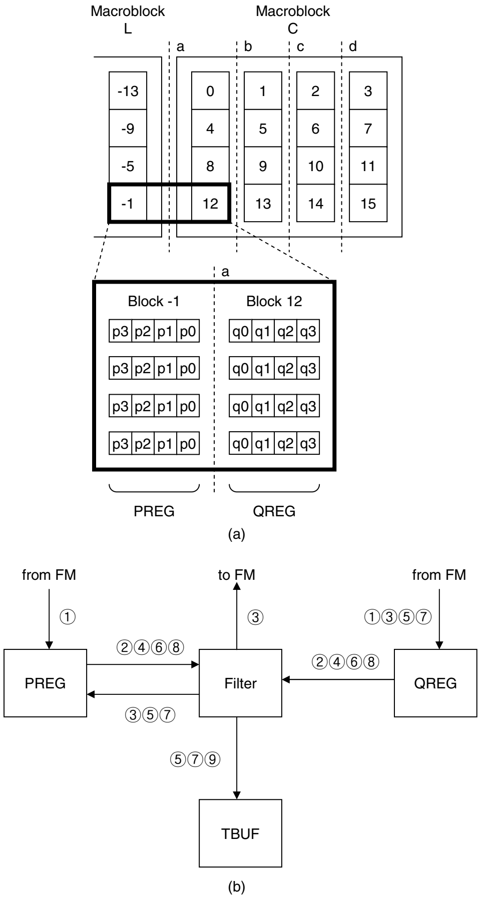


Fig. 8. Horizontal filtering. (a) Order of block processing. (b) Data flow.

same time, block 3 is read from FM to QREG for next block filtering (⑦). When (block 2, block 3) are processed, PREG and QREG are filtered (⑧). Filtered outputs of block 2 and 3 are stored in TBUT for vertical filtering (⑨). Thus block 0 to block 3 are finished. Block 4 to block 15 are processed in the same way with block 0 to 3.

Fig. 9 illustrates vertical filtering. Current macroblock C has four horizontal edges a, b, c, and d to be processed, as shown in Fig. 9 (a). Pixel data of upper macroblock U are required to process horizontal edge a. 4×4 blocks are filtered from block 0 to block 15, i.e. (block -13, block 0), (block 0, block 1), (block 1, block 2), (block 2, block 3), (block -9, block 4), ... (block 14, block 15) are filtered sequentially.

Vertical filtering is illustrated in detail in Fig. 9 (b). When (block -13, block 0) are processed, block -13 and 0 are read from FM and TBUF to PREG and QREG, respectively (①). Then, PREG and QREG are filtered (②). Filtered outputs of block -13 are stored in FM (③). At the same time, filtered outputs of block 0 are transfered to PREG for next block filtering (③). At the same time,

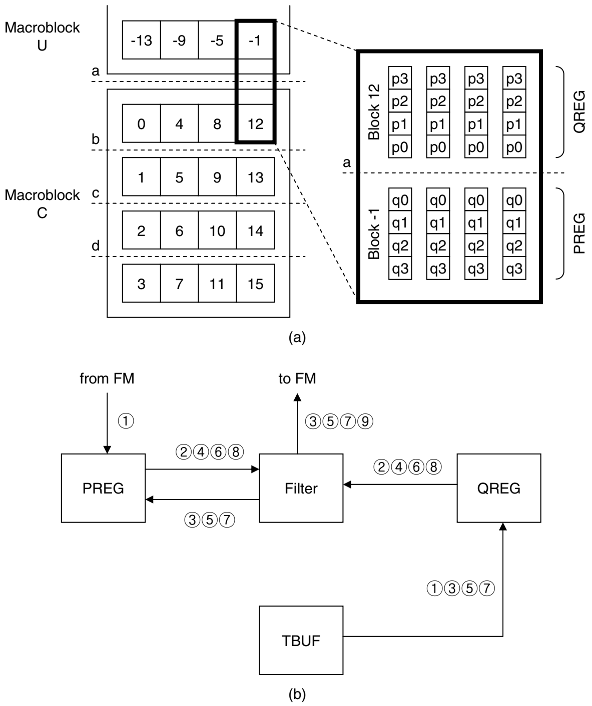


Fig. 9. Vertical filtering. (a) Order of block processing. (b) Data flow.

block 1 is read from TBUF to QREG for next block filtering (③). When (block 0, block 1) are processed, PREG and QREG are filtered (④). Filtered outputs of block 0 are stored in FM (⑤). At the same time, filtered outputs of block 1 are transfered to PREG for next block filtering (⑤). At the same time, block 2 is read from TBUF to QREG for next block filtering (⑤). When (block 1, block 2) are processed, PREG and QREG are filtered (⑥). Filtered outputs of block 1 are stored in FM (⑦). At the same time, filtered outputs of block 2 are transfered to PREG for next block filtering (⑦). At the same time, block 3 is read from TBUF to QREG for next block filtering (⑦). When (block 2, block 3) are processed, PREG and QREG are filtered (⑧). Filtered outputs of block 2 and 3 are stored in FM (⑨). Thus block 0 to block 3 are finished. Block 4 to block 15 are processed in the same way with block 0 to 3.

The proposed low-power H.264 was designed and implemented in a silicon chip. It was described in Verilog HDL. It was synthesized in 0.35 μm standard cell technology using Synopsys Design Compiler supported by IC Design Education Center. The total gate counts are about 20,000 gates including control circuit and deactivating circuit. The maximum operation frequency is 108 MHz. The maximum throughput is 30 frame/s with CCIR601 image format (704×576 pixels, 30 frame/s) at 108 MHz and 120 frame/s with QCIF image format (176×144 pixels, 30 frame/s) at 27 MHz. Fig. 10 shows the layout of the implemented chip.

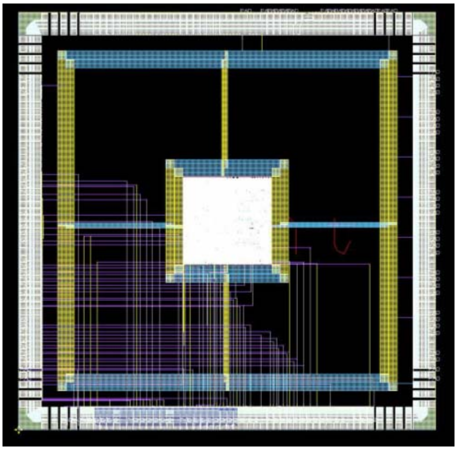


Fig. 10. Layout of the proposed low-power H.264 deblocking filter

4 Conclusion

In this paper, a low-power H.264 deblocking filter algorithm was proposed. In H.264 deblocking filter, filtering can be skipped on several pixels when some skipping conditions hold, and power consumption can be greatly reduced by totally or partially deactivating deblocking filter. Simulation results show that the power consumption is reduced up to 20.3% when compared to conventional algorithms. A low-power H.264 deblocking filter architecture was also proposed, and it was implemented in silicon chip. The gate count, the maximum operation frequency, and the maximum throughput are about 20,000 gates, 108 MHz, and 30 frame/s with CCIR601 image format, respectively.

References

1. ITU-T Rec. H.264: Advanced Video Coding for Generic Audio Visual Services (2005).
2. ISO/IEC JTC1/SC29/WG11 13818-1: Coding of Moving Pictures and Associated Audio: Video (1994).
3. ISO/IEC JTC1/SC29/WG11 14496-2: Coding of Audiovisual Object: Visual (1998).
4. Huang, Y., Chen, T., Hsieh, B., Wang, T., Chang, T., Chen, L: Architecture Design for Deblocking Filter in H.264/JVT/AVC, International Conference on Multimedia and Expo (2003) 693-696.
5. Rabaey, J.: Low-Power Silicon Architectures for Wireless Communication, Asia and South Pacific Design Automation Conference (2000) 379-380.
6. Jang, Y., Shin, Y., Hong, M., Wee, J., Lee, S.: Low-Power 32bit x 32bit Multiplier Design with Pipelined Block-Wise Shutdown, Lecture Notes in Computer Science **3769** (2005) 398-406.

Rate Control Scheme for Video Coding Using Low-Memory-Cost Look-Up Table

Linjian Mo, Chun Chen, Jiajun Bu[†], Xu Li, and Zhi Yang[⋆]

College of Computer Science, Zhejiang University, Hangzhou, P.R. China
[†]bjj@zju.edu.cn

Abstract. This paper presents a new MPEG-4/H.263 rate control scheme. It is applicable to constant-bit-rate transmissions containing equal-interval I-frames. Conventional schemes usually allocate bits equally to each P-frame, resulting in large video quality variations. In this work, in order to smooth quality variations, we allocate bits unequally to each P-frame by its encoding complexity and buffer level. In macroblock (MB) layer, rate-distortion model is usually employed to determine MB QP, but it needs complex floating-point computations. So, instead, we design a simple and effective look-up table, which also relates MB QP, SAD and the texture bit counts. With certain improvements, our table size reduces to only 1/20 of that in reference work, while the scheme still performs well. Experimental results show that, compared with a modified TMN8 rate control scheme, our scheme achieves higher PSNR and smoother PSNR variations, retaining the bit rate proximity and buffer regulation.

1 Introduction

Rate control plays an important role in video coding process. It is introduced to regulate and control the output bit rates of video sources in network to achieve the best trade-off between quality and bandwidth utilization [1]. In this work, rate control by adjusting quantization parameter (QP) in MPEG-4/H.263 is considered. We focus on constant-bit-rate transmissions containing equal-interval I-frames, that is, group-of-pictures (GOP) pattern is introduced. Instead of a low-delay (small) buffer, a larger buffer is needed to avoid frame skipping.

Usually, besides frame layer rate control, macroblock (MB) layer rate control is introduced to provide finer buffer regulations and higher bit rate encoding [2]. We propose a rate control scheme, including a new frame bit allocation algorithm and an improved MB layer rate control algorithm based on that of [3].

Frame bit allocation is performed in frame layer rate control. In many algorithms, such as [2] and [4], bits are allocated almost equally to each frame. Unfortunately, different frames have different encoding complexities, and an equal bit allocation may result in large video quality variations. We propose a new frame bit allocation algorithm, which considers frame encoding complexity, like [5]. Frames with higher encoding complexity are allocated with more bits, and

[⋆] Project supported by National Basic Research Program of China (2006CB303000).

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 780–790, 2006.

further adjustment by the buffer level should be performed. Our algorithm is simple and effective. It demonstrates that we achieve smoother PSNR variations, especially when scene changes or high motions occur.

After frame layer rate control, MB layer rate control is performed to determine MB QP. Conventional schemes usually employ rate-distortion models. One drawback is lots of complex floating-point computations are involved (other drawbacks can be found in [3]). Thus, similar to [3], we design a look-up table, which also relates MB QP, SAD and bit counts for encoding texture information (texture bit counts). Only some simple computations are involved. We further propose several improvements, resulting in a much smaller table (about 20 times smaller) than that of [3], while the scheme still has good performance. Meanwhile, the computing complexity is not increased.

Experimental results show that, compared with a modified TMN8 rate control scheme, our scheme achieves higher PSNR and smoother PSNR variation, while the average bit rate is close to the target, and few buffer overflows caused by the uncontrollable I-frames occur. Furthermore, similar to that of [3], our MB layer rate control algorithm is also applicable to low-delay mode of H.263.

The rest of the paper is organized as follows. Section 2 presents the proposed rate control scheme with the order, frame layer rate control, MB layer rate control and summarization. In Section 3, we show the experimental results to evaluate the performance of our scheme. Finally, conclusions are presented in Section 4.

2 Proposed Rate Control Scheme

2.1 Frame Layer Rate Control

Initialization. Assume each GOP has the same bit budget B_{GOP}. According to the channel bandwidth requirement, B_{GOP} is set to $N_{GOP} \times R/F$, where N_{GOP} is the number of frames in one GOP; R is the channel bit rate; F is the frame rate. Further, in order to be resilient to the buffer fluctuation, the encoder buffer should be larger than the low-delay (small) buffer (thus allocating bits to frames unequally is possible).

In this work, QP of the I-frame in each GOP is set to be the average QP of all frames in the last GOP. As the I-frame is the first frame in GOP, this method will reduce video quality variations, and ensure the quality of entitle GOP.

Typically, I-frame produces much more bits than P-frame, and the encoding buffer level will increase rapidly after the bits used for encoding an I-frame are put in. Thus, the GOP initial buffer level before the first I-frame comes should be low enough. Otherwise, the incoming I-frame will cause the buffer overflow. In this work, we set the initial buffer level L_{init} to 20% of the buffer size.

We describe the frame skipping control, which is performed in most schemes.

Frame Skipping Control. The frame skipping control is similar to that of [2] and [4]. We describe it briefly as follows. Before encoding the current frame, the actual encoder buffer level should be updated by

$$ABL_{cur} = \max\{ABL_{prev} + B_{act} - R/F, 0\} \tag{1}$$

where

- B_{act}: the actual number of bits used for encoding the previous frame;
- ABL_{prev}: the previous buffer level.

If ABL_{cur} is larger than a predefined threshold M (e.g. 80%×buffer size), the encoder skips frames until the buffer fullness is below M. For each skipped frame, buffer level is reduced by R/F bits. The maximum buffer delay is M/Rs.

After frame skipping control, we focus on how to better allocate bits to each P-Frame in a GOP.

P-frame Bit Allocation

(1) Allocate bits by P-frame encoding complexity Usually, in order to provide a better video quality and reduce the quality variations, we should allocate bits differently to each P-frame in one GOP, instead of allocating equally. Therefore, similar to [5], frame encoding complexity is considered. Frames with higher encoding complexity are allocated with more bits. In order to reducing the computing complexity, we simply use the frame SAD to indicate the frame encoding complexity. Our algorithm is proposed as follows.

According to the relation between MB SAD and the average texture bit counts showed in Fig.1, we find that when QP is small (e.g. QP<6), MB SAD and the texture bit counts approximately follow a power relation; when QP is large (e.g. QP≥6), it's approximately a linear relation. For simplicity, these relations are used in frame layer to do bit allocation. The formulas are

$$For\ small\ QP\ (QP < 6) : B_{tar\_p} = \overline{B_p} \times \sqrt{SAD_{cur\_p}/\overline{SAD_p}} \tag{2}$$

$$For\ large\ QP\ (QP \geq 6) : B_{tar\_p} = \overline{B_p} \times SAD_{cur\_p}/\overline{SAD_p} \tag{3}$$

where

- QP: QP of the current P-frame; simulated by the average QP of MBs in the previous frame;
- B_{tar_p}: the target number of bits allocated to the current P-frame;
- $\overline{B_p}$: the average number of bits used for encoding each P-frame in the GOP; defined by $(B_{GOP} - B_I)/(N_{GOP} - 1)$, where B_I is the number of bits already used for encoding the first I-frame;
- SAD_{cur_p}: SAD of the current P-frame;
- $\overline{SAD_p}$: the average SAD of P-frames in the GOP; simulated by the average SAD up to the current P-frame.

B_{tar_p} is further restricted in $[\overline{B_p}/2, 3 \times \overline{B_p}/2]$ to avoid large buffer variation.

(2) Adjust bits by the buffer level Only allocating bits by frame encoding complexity may result in large buffer variations, so we further adjust B_{tar_p} by the buffer level. A modified formula of the one in [2] is used. We describe it as follows.

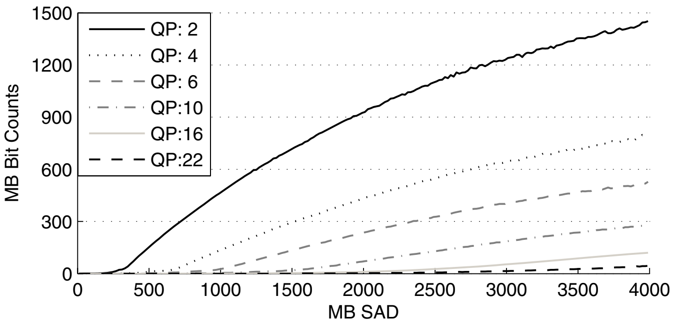


Fig. 1. The relation between MB SAD and the average texture bit counts (obtained by a H.263 encoder)

$$B'_{tar\_p} = B_{tar\_p} \times (4 \times TBL - ABL)/(2 \times TBL + ABL) \tag{4}$$

where TBL and ABL indicate the target buffer level and the actual buffer level.

TBL is defined as formula (5). It's a variable that would increase rapidly after bits of I-frame are put in, and decrease gracefully to L_{init} over again at the end of the GOP. This is different from [2]. In [2], the target buffer level is constant, which is not applicable to applications containing equal-interval I-frames [6].

$$\begin{cases} TBL_1 = L_{init} + B_I - R/F \\ TBL_{k+1} = TBL_k + \overline{B_p} - R/F \end{cases} \tag{5}$$

Furthermore, B'_{tar_p} should be larger than a lower bound (e.g. $\overline{B_p}/3$) to ensure the video quality.

Experimental results show that our algorithm results in smoother PSNR variations, especially when scene changes or high motions occur.

2.2 Macroblock Layer Rate Control

As bits are allocated unequally to each frame, it's better to perform MB layer rate control to avoid potential buffer overflow/underflow, and also for the high bit rate coding. We will describe how to allocate B'_{tar_p} to each MB and determine MB QP in this section.

Macroblock Bit Allocation. MB bit allocation is similar to that of [3]. To reduce the computing complexity, the residual MBs in one frame are categorized into compensable MBs with SAD$\leq T$ (a predefined threshold) and uncompensable MBs with SAD$> T$. The former does not need to be further encoded, and in the rest of the paper, MB(s) denotes the uncompensable MB(s) by default.

The target number of bits allocated to the kth MB for encoding texture information is defined as

$$b_{tar\_k} = B_{tex\_rem} \times SAD_k/(\sum_{i=k}^{N} SAD_i) \tag{6}$$

where

- B_{tex_rem}: the remaining number of bits used for encoding texture information of the remaining MBs;
- SAD_k: SAD of the kth MB;
- N: the total number of uncompensable MBs.

B_{tex_rem} is computed as

$$B_{tex\_rem} = B_{rem} - B_{uncode} - B_{header} - \overline{b_{h\_intra}} \times N_{intra} - \overline{b_{h\_inter}} \times N_{inter} \qquad (7)$$

where

- B_{rem}: the remaining number of bits used for encoding remaining MBs; initially, it is set to B'_{tar_p};
- B_{uncode}: the number of bits used for encoding partial non-texture information of compensable MBs;
- B_{header}: the number of bits used for encoding picture header information;
- $\overline{b_{h\_intra/h\_inter}}$: the average number of bits used for encoding intra/inter MB header information; simulated by the average value of all encoded intra/inter MBs;
- $N_{intra/inter}$: the number of remaining intra/inter MBs.

After MB bit allocation, we determine an appropriate MB QP. Instead of employing a mathematical rate-distortion model, we determine QP by a look-up table. Though the method is similar to [3], we propose several improvements to reduce the memory cost of the table, while still retain good performance. We will explain these improvements during the following descriptions.

Construction of the Look-Up Table. Usually, MB QP, its SAD and the bit counts for encoding texture information are used to set up rate-distortion models. Thus, we also choose these three parameters to construct a look-up table. This QP-SAD-bits table is indicated by a 2-D matrix b[QP][SAD_SEG_NO] (SAD_SEG_NO stands for SAD segment number). By a certain MB QP and SAD_SEG_NO, we can obtain the texture bit counts b.

The table is constructed off-line with the following process.

(1) For a certain QP, we encode a large number of video test sequences by the encoder that our scheme will be implemented in. For MBs with the same SAD, we compute the average texture bit counts. After this process, the relation between SAD and texture bit counts are set up. The curve in Fig.2 illustrates one such relation. We find that SAD-bits curve can be simulate by a piecewise linear function (illustrated in Fig.2). Thus, in order to reduce the memory cost, we choose SADs by a constant step SAD_STEP (e.g. 50 in our experiment, and the chosen SADs are 0, 50, 100 . . .), instead of continuous choosing (0, 1, 2 . . .) in [3]. The second parameter in the matrix indicates the SAD segment number SAD_SEG_NO (SAD_SEG_NO=⌊SAD/SAD_STEP⌋, where ⌊ ⌋ means the *floor* operation), and b[QP][SAD_SEG_NO] indicates the average texture bit counts of MBs with SAD=SAD_SEG_NO×SAD_STEP. Within each SAD segment, the relation between SAD and texture bit counts is simulated by a linear function, so recording the two end points of each segment is enough. Furthermore, we enlarge the SAD range to 4000, instead of 1660 in [3]. A larger SAD range covers more MBs for bit estimation.

(2) We do operations in *(1)* for all QPs to construct the look-up table.

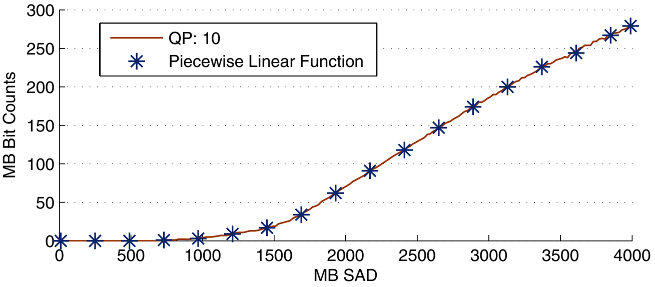


Fig. 2. The relation between MB SAD and the average texture bit counts with QP = 10 (obtained by a H.263 encoder)

Determination of MB QP. We consider the kth MB. After getting its SAD$_k$ and b_{tar_k}, we can determine an appropriate QP by the QP-SAD-bits table. The process is described as follows.

(1) Determine the QP search range.

To avoid large quality variations between adjacent frames, the QP search range is restricted, instead of the range [1, 31] used in [3]. Assume the average QP of MBs in the previous P-frame is QP$_{prev}$. The QP search range of MBs in the current P-frame is restricted to $[max\{\mathrm{QP}_{prev} - 2, 1\}, min\{\mathrm{QP}_{prev} + 2, 31\}]$. Noticed that if the current P-frame is the first P-frame in the sequence, the QP search range is still [1, 31].

By this restriction, the search complexity is rapidly reduced.

(2) Determine the SAD segment.

SAD$_k$ of the kth MB is located in the SAD segment [SAD$_m$, SAD$_n$], where SAD$_m$=SAD_STEP×($\lfloor$SAD$_k$/SAD_STEP$\rfloor$) and SAD$_n$=SAD$_m$+SAD_STEP.

(3) Estimate the texture bit counts for each QP in the search range.

For each QP in the search range, we estimate the texture bit counts b_{QP}. Instead of getting b_{QP} directly from the table in [3], we do interpolation to get b_{QP}. The formula is

$$b_{QP} = (b_n - b_m) \times (SAD_k - SAD_m)/(SAD\_STEP + b_m) \tag{8}$$

where b_m is b[QP][SAD$_m$/SAD_STEP] and b_n is b[QP][SAD$_n$/SAD_STEP].

(4) Choose an appropriate QP.

After getting b_{QP} for each QP in the range, we choose an appropriate QP, whose b_{QP} has a minimum distance to $b_{tar\_k}(min\{|b_{QP} - b_{tar\_k}|\})$.

(5) Adjust the chosen QP.

The QP chosen in *(4)* should be adjusted further. For example, in MPEG-4/H.263, the QP difference of horizontal adjacent MBs is restricted in [-2, 2].

Up to now, MB QP is determined and is used for quantization.

Updating of the Look-Up Table. After encoding each MB, we record the updating value. But the actual updating of the look-up table is performed after encoding the entire frame, instead of after encoding each MB in [3]. One reason is that lower updating frequency results in lower computing complexity; the other reason is it leads to more accurate updating. The process is described as follows.

We construct two extra tables. One is the updating table, which is used to record the updating values, defined as Δb[QP][SAD_SEG_NO]. The other is the counting table, which is used for counting, defined as C[QP][SAD_SEG_NO]. They are all initialized to 0.

In practice, the size of these two tables can be largely decreased. Considering the maximum possible updating range, the size of each table can be $5\times(\lfloor \text{SAD}_{max}/\text{SAD\_STEP}\rfloor - \lfloor \text{SAD}_{min}/\text{SAD\_STEP}\rfloor+2)$, where 5 is the number of all possible QPs in our search range; SAD_{max} and SAD_{min} indicate the maximum and minimum MB SAD in the frame; $(\lfloor \text{SAD}_{max}/\text{SAD\_STEP}\rfloor - \lfloor \text{SAD}_{min}/\text{SAD\_STEP}\rfloor+2)$ indicates the number of all possible updating points for a certain QP.

For the kth MB with SAD_k, after encoding it under the chosen QP, the actual texture bit counts is b_{act_k}. The texture bit counts we estimate by interpolating the look-up table is b_{QP}. Thus, the updating value is simply defined as $u = (b_{act\_k} - b_{QP})/2$. We record the updating value as follows.

```
(Assume SAD$_k$ is located in the SAD segment [SAD$_m$, SAD$_n$])
IF (SAD$_k$-SAD$_m$<SAD_STEP/4) THEN
  $\Delta$b[QP][SAD$_m$/SAD_STEP]+=u
  ++C[QP][SAD$_m$/SAD_STEP]
ELSE IF (SAD$_n$-SAD$_k$<SAD_STEP/4) THEN
  $\Delta$b[QP][SAD$_n$/SAD_STEP]+=u
  ++C[QP][SAD$_n$/SAD_STEP]
ELSE
  $\Delta$b[QP][SAD$_m$/SAD_STEP]+=u
  $\Delta$b[QP][SAD$_n$/SAD_STEP]+=u
  ++C[QP][SAD$_m$/SAD_STEP]
  ++C[QP][SAD$_n$/SAD_STEP]
```

We do the recording above for each MB once encoded. After encoding the entire frame, for those elements in the updating table with values changed (the corresponding C[QP][SAD_SEG_NO] is nonzero), Δb[QP][SAD_SEG_NO] is recomputed by

$$\Delta b[QP][SAD\_SEG\_NO]/ = C[QP][SAD\_SEG\_NO] \tag{9}$$

The matrix in Fig.3(a) indicates a sample of the updating table. Different rows mean different QPs, and different columns mean different SAD_SEG_NOs. Elements with values changed are marked by $\Delta b1$, $\Delta b2$, ... For other elements in the same columns with $\Delta b1$, $\Delta b2$, ..., we also change values as Fig.3(b) showed. It is called correlative changes.

After correlative changes, the final updating table is got, and it's used to update the corresponding elements in the look-up table. The formula is

$$b[QP][SAD\_SEG\_NO]+ = \Delta b[QP][SAD\_SEG\_NO] \tag{10}$$

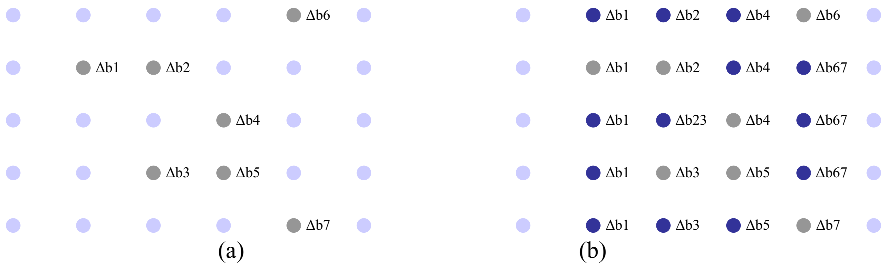


Fig. 3. (a)A sample of the updating table after computations using formula (9). (b)A part of the updating table after correlative changes (The figure illustrates different types of correlative changes; $\Delta b23 = (\Delta b2 + \Delta b3)/2, \Delta b67 = (\Delta b6 + \Delta b7)/2.$).

2.3 Summarization

The proposed rate control algorithm is summarized as follows.

Off-line: Construct a QP-SAD-bits look-up table (see Subsection 2.2).
On-line: For each P-frame:
- **Step 1:** Do frame skipping control (see Subsection 2.1). If the current frame is not skipped, go to Step 2. Otherwise, continue for the next frame.
- **Step 2:** Perform motion estimation and compensation for all MBs. Record SAD and motion vector of each MB, and categorize the residue MBs into compensable and uncompensable types. The uncompensable MBs are further classified into intra and inter mode [3].
- **Step 3:** Compute the frame target bit counts B'_{tar_p} (see Subsection 2.1).
- **Step 4:** Prepare for MB bit allocations: calculate B_{uncode}, B_{header}, N_{intra} and N_{inter}; set $B_{rem} = B'_{tar\_p}$, $\overline{b_{h\_intra}} = 0$ and $\overline{b_{h\_inter}} = 0$.
- **Step 5:** For each uncompensable MB (assume the current MB is the kth MB):
  - **Step 5.1:** Calculate B_{tex_rem} using (7).
  - **Step 5.2:** Calculate the target texture bit counts b_{tar_k} using (6).
  - **Step 5.3:** Determine the MB QP (see Subsection 2.2).
  - **Step 5.4:** Encode the MB.
  - **Step 5.5:** Calculate the actual texture bit counts b_{act_k}. Record the updating value by the pseudocode in Subsection 2.2.
  - **Step 5.6:** Calculate the actual header bit counts b_{hd_k}. Update $\overline{b_{h\_intra}}$ or $\overline{b_{h\_inter}}$.
  - **Step 5.7:** Update N_{intra} or N_{inter}.
  - **Step 5.8:** Update B_{rem} by $B_{rem} -= (b_{act\_k} + b_{hd\_k})$.
- **Step 6:** Update the QP-SAD-bits look-up table using (10).

From the viewport of hardware implementation, our algorithm is also much cheaper than the existing mathematical R-D model based rate control algorithm. The hardware implementation is similar to that of [3], and we omit it here.

3 Experimental Results

For the convenience of our comparisons, we implement the proposed rate control scheme in UBC H.263 encoder (version 3.0) [7]. In our experiments, the encoder

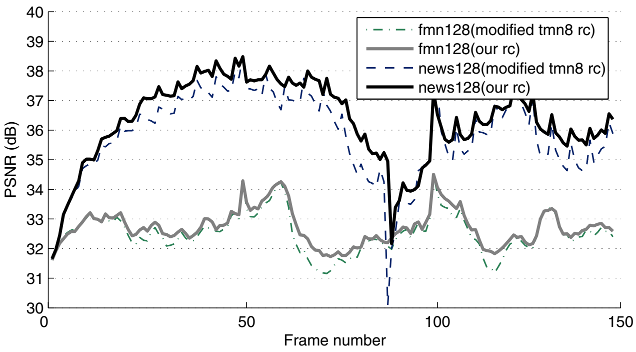


Fig. 4. Comparison of PSNR performance (GOP size: 50f, R: 128Kbps, F: 30fps)

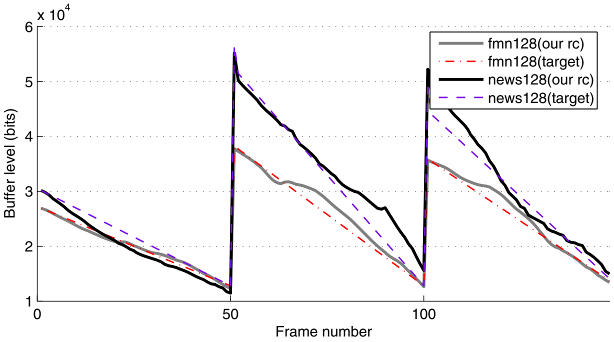


Fig. 5. Encoder Buffer level variations of our scheme (GOP size: 50f, R: 128Kbps, F: 30fps, Buffer size: 64K)

buffer size is set to $R \times 0.5$ instead of a low-delay (small) buffer size; this results in a maximum buffer delay 500ms; an I-frame appears every 50 frames (GOP size: 50f); the first I-frame is coded with QP=13. For fair comparisons, we uniformly divide a rate control scheme into frame layer and MB layer rate control. Experiments are done as follows. In MB layer rate control, similar to [3], we compare ours with the well-known TMN8 MB layer rate control algorithm [4], [8]; but in frame layer rate control, for the comparison of frame bit allocation algorithms, we make some changes.

The frame bit allocation algorithm in TMN8 rate control scheme aims at low-delay communications, which is not appropriate in our application environment. Thus, it is unfair to compare our frame bit allocation algorithm with that of TMN8. Instead, in order to evaluate our algorithm with the consideration of frame encoding complexity, we compare ours (formulas (2)-(5) are all used) with the algorithm without considering frame encoding complexity (bits are allocated equally to each P-frame with $\overline{B_p}$, and only formulas (4)-(5) are used).

Therefore, our scheme is compared with a modified TMN8 rate control scheme. We describe it in Table 1.

Table 1. Comparisons of our RC scheme and the modified TMN8 RC scheme

	Our rc scheme	**Modified TMN8 rc scheme**
Frame layer	Formulas (2)-(5) used	Equal bit allocation; formulas (4)-(5) used
MB layer	Look-up table based rc	TMN8 MB layer rc

Table 2. Comparisons of the average bit rate

Test Name	**Video Sequences**	**Fame rate fps**	**Target rate Kbps**	**Modified TMN8 rc**	**Our rc**
fmn80	"foreman.qcif"	30	80	80.02	79.97
fmn128	"foreman.qcif"	30	128	128.01	128.02
mad80	"mthr-dotr.qcif"	30	80	79.97	79.90
mad128	"mthr-dotr.qcif"	30	128	127.96	127.88
news80	"news.qcif"	30	80	80.17	80.20
news128	"news.qcif"	30	128	128.26	128.39
suzie80	"suzie.qcif"	30	80	80.00	79.86
suzie128	"suzie.qcif"	30	128	127.99	127.90

Table 3. Comparisons of the number of skipped frames and the average PSNR

Test Names	**Total Frames**	**Modified TMN8 rc #Frames Skipped**	**Our rc #Frames Skipped**	**Modified TMN8 rc PSNR dB**	**Our rc PSNR dB**	**Gain in PSNR dB**
fmn80	150	0	0	30.70	30.84	0.14
fmn128	150	0	0	32.51	32.71	0.20
mad80	150	0	0	34.08	34.26	0.18
mad128	150	0	0	35.89	36.11	0.22
news80	150	4	4	33.16	33.52	0.36
news128	150	2	2	35.82	36.35	0.53
suzie80	150	0	0	35.35	35.50	0.15
suzie128	150	0	0	37.00	37.22	0.22

Some experimental results are showed in Fig.4, 5 and Table 2, 3. Fig.4 shows comparisons of PSNR performance. It can be seen that we achieve higher PSNR for most of the time; also, we achieve smoother PSNR variations, especially when scene changes or high motions occur. For example, scene changes happen during the 89^{th} to 92^{nd} frames in "news128" sequence, and PSNR of the modified TMN8 scheme drops 4.2 dB, whereas our scheme only drops 2.8 dB. In Fig.5, we compare variations of our encoder buffer level with the target buffer level. We can see that, though bits are allocated to each P-frame unequally, our buffer level is close to the target buffer level as a whole. Table 2 and 3 show comparisons of the average bit rate, PSNR and the number of skipped frames for different sequences. It can be seen that we achieve better average PSNR than the modified TMN8 scheme; also, the average bit rate is close to the target, and few buffer overflows caused by the uncontrollable I-frames occur.

4 Conclusions

In this paper, a new rate control scheme is proposed, which is applicable to constant-bit-rate transmissions containing equal-interval I-frames (GOP pattern). For frame bit allocation, we take frame encoding complexity and buffer level into consideration, resulting in an unequal bit allocation to each P-frame, and smoother video quality variations. In MB layer rate control, similar to [3], we design a look-up table that also relates MB QP, SAD and the texture bit counts. With several improvements, our table size reduces to only 1/20 of the one in [3], while the scheme still has good performance, with nearly no computing complexity increasing. Experimental results show that, compared with a modified TMN8 rate control scheme, our scheme achieves higher PSNR and smoother PSNR variations, retaining the bit rate proximity and buffer regulation.

References

1. Girod, B.: Scalable Video for Multimedia Systems, Computers & Graphics, Vol. 17, (1993), 269–276
2. Lee, H.J., Chiang, T., Zhang, Y.Q.: Scalable Rate Control for MPEG-4 Video, IEEE Trans. Circuits Syst. Video Technol., Vol. 10, (2000), 878–894
3. Tsai, J.Ch.: Rate Control for Low-Delay Video Using a Dynamic Rate Table, IEEE Trans. Circuits Syst. Video Technol., Vol. 15, (2005), 133–137
4. Ribas-Corbera, J., Lei, S.: Rate Control in DCT Video Coding for Low-delay Video Communication, IEEE Trans. Circuits Syst. Video Technol., Vol. 9, (1999), 172–185
5. Jiang, M.Q., Yi, X.Q. and Ling, N.: Improved Frame-layer Rate Control for H.264 Using MAD Ratio, IEEE International Symposium on Circuits and Systems, Vol. 3, (2004), 813–816
6. Pan, F., Li, Z., Lim, K., Feng, G.: A Study of MPEG-4 Rate Control Scheme and Its Improvements, IEEE Trans. Circuits Syst. Video Technol., Vol. 13, (2003), 440–446
7. UBC H.263 codec: ITU-T/SG15, Video codec test model, TMN8, University of British Columbia, Canada, (1997)
8. ITU-T/SG15: Video codec test model, TMN8, Portland, June, (1997)

Adaptive GOP Bit Allocation to Provide Seamless Video Streaming in Vertical Handoff

Dinh Trieu Duong, Hye-Soo Kim, Le Thanh Ha,
Jae-Yun Jeong, and Sung-Jea Ko

Department of Electronics Engineering, Korea University,
Anam-Dong Sungbuk-Ku, Seoul, Korea
{duongdt, hyesoo, ltha, jyjeong, sjko}@dali.korea.ac.kr

Abstract. Vertical handoff is required to offer users the capability of achieving anywhere and anytime internet access with the full service support from interoperability between third generation (3G) and wireless LAN (WLAN) network. However, video data can be lost due to latency caused by vertical handoff. To solve this problem, in this paper, we propose an efficient GOP bit allocation method to provide seamless video streaming in vertical handoff. In the proposed method, the streaming server first predicts the network status by using the wireless channel modeling and then adaptively performs bit allocation for a group of pictures (GOP) to adapt video delivery to vertical handoff scenario. Experimental results show that the proposed method can provide significant improvement to video streaming performance, in terms of users-perceive video quality, and robustness against the latency and packet loss caused by vertical handoff.

Keywords: Bit allocation, Vertical handoff, GOP, Rate control.

1 Introduction

It is now commonly understood that the fourth generation (4G) network will comprise a variety of wireless access networks in a complementary manner. The 4G wireless networks will integrate different types of networks such as WLAN and 3G network because no single wireless network technology simultaneously can provide a low latency, high bandwidth, and wide area data service to a large number of mobile users.

The movement of a user within or among different types of networks is called the vertical mobility. One of the major challenges for seamless service in the vertical mobility is the vertical handoff, where handoff is the process of maintaining a mobile user's active connection by changing its point of attachment [1]. In the 4G wireless systems, the vertical handoff with small latency and packet losses is one of the key factors to provide the seamless service for multimedia communication with real-time requirements.

In vertical handoff, the seamless video streaming is taken great care from users. It is defined as the video streaming that provides seamless playout at the client during vertical handoff. In video communication, video data can be lost due to the latency caused by vertical handoff. The video quality degradation due to the packet loss and

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 791 – 801, 2006.

latency in vertical handoff is the critical problem in seamless video streaming. To solve this problem, a successful video streaming solution needs to adapt appropriately to vertical handoff scenarios.

There are several methods to achieve seamless video streaming in vertical handoff [2]-[4]. The multimedia transport protocol (MMTP) determines the encoding rate according to the measured available bandwidth in vertical handoff [2]. However, this protocol does not concern the packet loss caused by vertical handoff. This would result in an inefficient solution to maintain a seamless streaming with high video quality during vertical handoff. The method in [3] presents a new scheme to achieve negligible delay, maximum throughput with low packet loss ratio for optimal handoff process between WLAN and general packet radio service (GPRS) network. However, this scheme requires the support of the system-level design such as the hardware configuration and the lower layer protocol design. The QoS based vertical handoff in [4] uses QoS supportable access points (APs) in order to achieve seamless handoff between the universal mobile telecommunications system (UMTS) and the WLAN network. However, it also needs the system-level design.

In this paper, we propose an adaptive GOP bit allocation (A-GBA) method to provide seamless video streaming in vertical handoff between the WLAN and 3G network. The proposed method can efficiently perform in the encoder of streaming server without the system-level design. In order to achieve seamless video streaming, the streaming server first predicts the network status and estimates the channel rate in vertical handoff by using a three-state Markov channel modeling. Then, based on the estimated channel rate and the packet loss caused by vertical handoff, the streaming server adaptively performs the proposed A-GBA method to adapt the video stream to the current channel properties. Since the key factors of vertical handoff such as the latency and packet loss are considered in the A-GBA method, the target number of bits of the GOP is adaptively allocated to maintain high video quality for seamless streaming in vertical handoff.

The paper is organized as follows. In the next section, we describe the wireless channel model for vertical handoff. The proposed A-GBA bit allocation is presented in Section 3. Section 4 shows the experimental results and our conclusions are given in Section 5.

2 Wireless Channel Model for Vertical Handoff

In order to reduce the video quality degradation in vertical handoff, we first specify the handoff procedure and then define an appropriate model for wireless channel in vertical handoff scenario.

Fig. 1 illustrates the vertical handoff procedure in the heterogeneous network where significant events have been pointed out. As shown in Fig. 1, T_I , T_V , and T_F , respectively, are the period for vertical handoff initiation, the handoff latency, and the period for the packet retransmission. During vertical handoff, the trigger messages indicating the handoff status are transmitted to the server by the client. These trigger messages are generated by using the receive signal strength indicator (RSSI) and the pilot signal strength (Ec/Io) in WLAN and 3G network respectively.

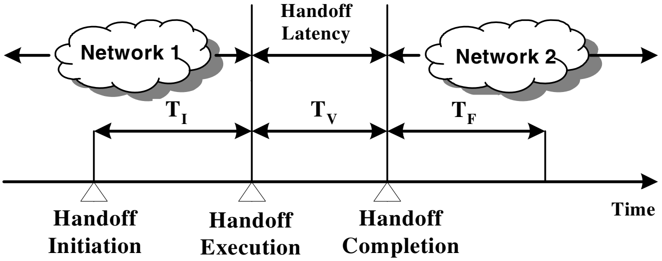


Fig. 1. Handoff procedure

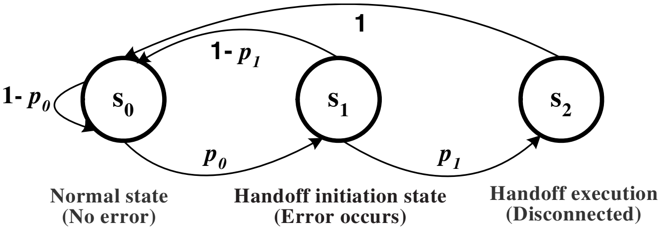

1
1- p_1
1- p_0
s_0
s_1
s_2
p_0
p_1
Normal state (No error)
Handoff initiation state (Error occurs)
Handoff execution (Disconnected)

Fig. 2. Three-state Markov model

In this paper, the wireless channel is modeled as the three-state Markov model. Fig. 2 shows the three-state Markov model for vertical handoff. This Markov model has three channel states, s_0, s_1, and s_2 where s_0, s_1, and s_2, respectively, are the "normal state", the "handoff initiation state", and the "handoff execution state". The transition probabilities can be obtained by using the channel characteristic information such as the RSSI and the Ec/Io measured in our experimental platform. When the channel is in state $s_n, n = \{0,1,2\}$, the transition of the channel state goes to the next higher state or back to state s_0 based on the channel information. If the channel is in state s_2, it will always transit to state s_0.

The transition probability matrix for the three-state Markov model can be set up as

$$\mathbf{P} = \begin{bmatrix} 1\text{-}\ p_0 & p_0 & 0 \\ 1\text{-}\ p_1 & 0 & p_1 \\ 1 & 0 & 0 \end{bmatrix}. \qquad (1)$$

In the Markov model, the vector of state probabilities $\boldsymbol{\pi}(k \mid S(t))$ at time k can be derived from the state probabilities $\boldsymbol{\pi}(k-1 \mid S(t))$ at the previous time slot and the transition probability matrix $\mathbf{P}$ in (1) as

$$\boldsymbol{\pi}(k \mid S(t)) = \boldsymbol{\pi}(k-1 \mid S(t)) \cdot \mathbf{P}. \qquad (2)$$

The vector of state probabilities at time k can be obtained by using equation (2) recursively as

$$\boldsymbol{\pi}(k \mid S(t)) = \boldsymbol{\pi}(t \mid S(t)) \cdot \mathbf{P}^{k-t}. \qquad (3)$$

In our channel model, packets are transmitted correctly when the channel is in state s_0, while errors occur when the channel is in states s_1 and s_2. Therefore, $\pi_0(k \mid S(t))$ is the probability of correct transmission at time *k,* and at the given channel state $S(t)$, the channel rate $\hat{R}$ can be estimated as

$$\hat{R} = E\left[C(k) \mid S(t)\right] = R_{max} \cdot \pi_0(k \mid S(t)), \tag{4}$$

where R_{max} is the maximum channel rate in WLAN or 3G network.

We consider the vertical handoff scenario between the WLAN and 3G network, where each network provides the different data rates. Thus, the estimated channel rate $\hat{R}$ is an important factor for the adaptive bit allocation method. Based on $\hat{R}$, we can efficiently allocate the target number of bits to each GOP as well as each frame in the GOP.

3 Proposed A-GBA Method

Bit allocation is necessary for video encoding to develop a proper rate control method. In bit allocation methods, a given bit budget of video source is efficiently assigned to the basic units of video encoding such as GOPs, frames, and macroblocks. Based on the target number of bits of the basic units obtained from bit allocation, the optimal rate control method is applied to achieve the best video quality at a given bitrate [5]-[6].

In this paper, the proposed A-GBA method includes the coarse GOP bit allocation (C-GBA) and the fine GOP bit allocation (F-GBA) method. The combination of the C-GBA and F-GBA methods enables the streaming server to perform the bit allocation efficiently for seamless video streaming in vertical handoff.

3.1 Coarse GOP Bit Allocation (C-GBA) Method

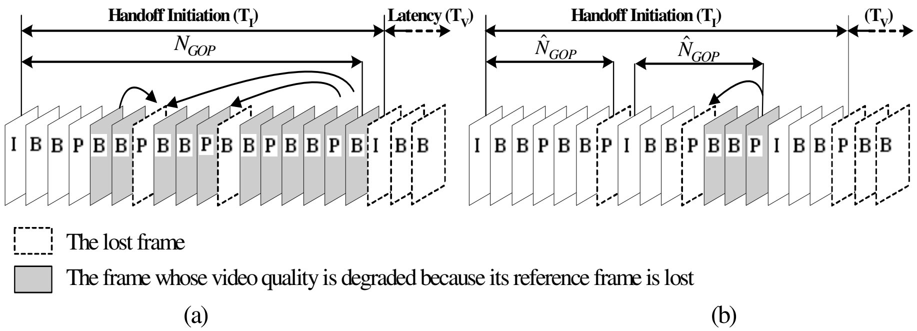
(a) N_{GOP} =17 (b) $\hat{N}_{GOP}$ =7

The structure of the GOP is composed of one I frame and some P and B frames. Let N_{GOP} denote the length of the GOP that is the total number of frames in the GOP. Let N_P and N_B be the numbers of P and B frame in the GOP, respectively. The number of bits allocated to the GOP, B_{GOP}, is defined as follows:

$$B_{GOP} = \left(1 + N_P + N_B\right) \cdot \frac{\hat{R}}{F} = N_{GOP} \cdot \frac{\hat{R}}{F}, \tag{5}$$

where F is the frame rate and $\hat{R}$ is the channel rate estimated in equation (4).

In vertical handoff, during the handoff initiation period, T_I, video data can be lost before a mobile station (MS) performs vertical handoff. This results in the degradation of the quality in video stream during T_I. In order to solve this problem,


Handoff Initiation (T_I)
Latency (T_V)
N_GOP
I B B P B B P B B P B B P B B P B I B B
Handoff Initiation (T_I)
(T_V)
N̂_GOP
N̂_GOP
I B B P B B P I B B P B B P I B B P B B
The lost frame
The frame whose video quality is degraded because its reference frame is lost

(a) (b)

Fig. 3. Degradation of video quality in the GOP when some reference frames are lost

we propose the C-GBA method, which appropriately reduces the length of theoriginal GOP, N_{GOP} , to produce the higher number of new GOPs during T_I . Using the proposed method, the target number of bits assigned to the new GOP is roughly estimated to adapt video stream to the current channel in the initiation period.

Fig. 3 illustrates the concept of the C-GBA method. As shown in Fig. 3 (a), when the length of the original GOP is long and the reference frame, especially the Intra-frame I, in its own GOP is lost during period T_I , the error propagation and the video quality degradation over frames of the GOP are considerable. Fig. 3(b) indicates the solution using the proposed C-GBA method. As shown in Fig. 3(b), since the number of GOPs and Intra-frames is increased in T_I , the quality degradation within the GOP is soon to be recovered in the new GOPs to maintain the high video quality for seamless streaming.

During T_I , when the length of the original GOP is reduced, the numbers of P and B frame of the GOP are adaptively changed to allocate the new target number of bits to the new GOP. Thus, let β denote the adaptive control factor of N_{GOP} , $\hat{N}_P$ and $\hat{N}_B$ denote the new numbers of P and B frame in the new GOP, respectively. $\hat{N}_P$ and $\hat{N}_B$ are adaptively controlled by

$$1 + \hat{N}_P + \hat{N}_B = \beta \cdot N_{GOP} \tag{6}$$

Fig.4 shows the controlling flowchart for the adaptive factor β . As shown in Fig.4, based on the information of the channel available bandwidth (BW) and the channel condition which is transmitted from the streaming client, the streaming server adaptively controls the value of β so as to achieve the acceptable video quality playing at the client during T_I . In the Fig.4, we assume that the most compact GOP structure is defined as IBBP with N_{GOP_Min} =4, then the initial value β_o of adaptive factor is initially set in the range of (N_{GOP_Min} / N_{GOP} , 1).

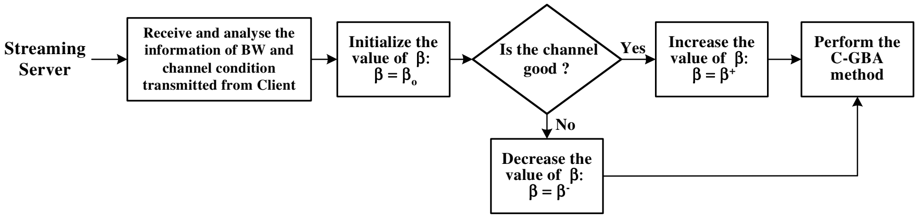

Streaming Server
Receive and analyse the information of BW and channel condition transmitted from Client
Initialize the value of β: β = β₀
Is the channel good ?
Yes
No
Increase the value of β: β = β⁺
Perform the C-GBA method
Decrease the value of β: β = β⁻

Fig. 4. Controlling flowchart for the adaptive factor

However, together with the increase in the number of Intra-frames, the encoding bitrate is also increased since a great number of bits are generated for Intra frame encoding. Therefore, to adapt to the current link channel where the available bandwidth is limited, $\hat{N}_P$ and $\hat{N}_B$ should satisfy the condition as follows:

$$\sum_{i=1}^{[n]}\left(\overline{B}_{I,i}+\hat{N}_P\overline{B}_{P,i}+\hat{N}_B\overline{B}_{B,i}\right)\leq T_I\cdot\hat{R}, \tag{7}$$

where $n=(T_I\cdot F/(1+\hat{N}_P+\hat{N}_B))$, $\left[n\right]$ represents an integer part of n , which is the total number of the new GOPs transmitted in the period T_I , $\overline{B}_{I,i}$, $\overline{B}_{P,i}$ and $\overline{B}_{B,i}$ are the average number of bits per frame of I, P, and B frame in the i^{th} new GOP , respectively.

Based on the new numbers of P and B frame, the target number of bits allocated to the new GOP is modified in the C-GBA method as

$$\hat{B}_{GOP}=\beta\cdot N_{GOP}\cdot\frac{\hat{R}}{F}=\left(1+\hat{N}_P+\hat{N}_B\right)\cdot\frac{\hat{R}}{F}\cdot \tag{8}$$

3.2 Fine GOP Bit Allocation (F-GBA) Method

The F-GBA method is performed after the C-GBA method to avoid the latency and packet loss caused in the handoff execution period, T_V . During T_V , the video quality degradation due to the packet loss and latency is the critical problem in seamless video streaming. As the solution to this problem, the streaming server in the F-GBA method is speeded up during the period T_I to transmits in advance all the frames that can be lost due to the latency in vertical handoff. Then, the streaming server adaptively performs the bit allocation to adapt video stream to the handoff execution scenario. In the F-GBA method, the target number of bits assigned to each frame of the GOP is calculated in detail to maintain high video quality during the period T_V .

Let F' and α denote the new frame rate and the speed factor of the original frame rate, F, respectively. To transmit in advance all the frames in the period T_I , α should satisfy the condition as follows:

$$\alpha=\frac{F'}{F}\geq\frac{\left(T_I+T_V\right)}{T_I}. \tag{9}$$

However, the picture quality can be degraded since all the frames cannot be transmitted in advance because of the channel bandwidth and buffer constraints. Therefore, during T_I the tradeoff between the speed factor α and the target number of bits allocated to each frame in the GOP should be considered to keep available bandwidth unchanged while maintaining the high video quality. To this end, in the proposed F-GBA method, we first redefine the target number of bits $\hat{B}_{GOP}$ obtained in the C-GBA method as the function of α as follows:

$$\hat{B}_{GOP} = \left(1 + \hat{N}_P + \hat{N}_B\right) \cdot \frac{\hat{R}}{\alpha.F}. \tag{10}$$

Then, we apply the TM5 model [7] to efficiently allocate the target number of bits to each frame in the GOP. Using the TM5 model, the number of bits assigned to the remaining frame in the GOP is defined by

$$B_{R,j} = \begin{cases} \hat{B}_{GOP} & j\text{=1} \\ B_{R,j-1} - \dfrac{B_{j-1}}{\alpha} & j\text{=2, 3,... } \hat{N}_{GOP} \end{cases} \tag{11}$$

where B_{j-1} is the number of encoding bits generated to the *(j-1)*th frame in the GOP.

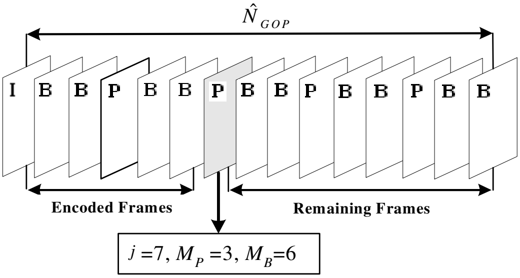


Fig. 5. An example of remaining frames in the GOP

After encoding a frame, the number of bits assigned to remaining frames is updated by using equation (11). Fig. 5 shows an example of the remaining frames in a GOP, where M_P and M_B are the number of P and B frames remaining in the GOP.

The complexity of a frame is defined as

$$C = B .\overline{Q}, \tag{12}$$

where B is the number of encoding bits generated to the frame and $\overline{Q}$ is the average quantization parameter of the frame. Since the complexity is different among the frames in the GOP, F-GBA method defines the weight factors corresponding to the complexity of I, B and P frame as follows:

$$W_I = \frac{C_I}{C_I + M_P.C_P + M_B.C_B}, \; W_P = \frac{C_P}{M_P.C_P + M_B.C_B} \text{ and } W_P = \frac{C_B}{M_P.C_P + M_B.C_B},$$

where $C_I = B_I.\overline{Q_I}$, $C_P = B_P.\overline{Q_P}$ and $C_B = B_B.\overline{Q_B}$ as defined in equation (12), respectively.

Based on $B_{R,j}$ and frame complexity, the target number of bits assigned to I, P and B frame in the GOP is given by

$$Tar_I = \max(B_{R,1}.W_I, \frac{\hat{R}}{8.F}) = \max(\frac{B_{R,1}.C_I}{C_I + M_P.C_P + M_B.C_B}, \frac{\hat{R}}{8.F}). \tag{13}$$

$$Tar_P = \max(B_{R,j}.W_P, \frac{\hat{R}}{8.F}) = \max(\frac{B_{R,j}.C_P}{M_P.C_P + M_B.C_B}, \frac{\hat{R}}{8.F}). \tag{14}$$

$$Tar_B = \max(B_{R,j}.W_B, \frac{\hat{R}}{8.F}) = \max(\frac{B_{R,j}.C_B}{M_P.C_P + M_B.C_B}, \frac{\hat{R}}{8.F}). \tag{15}$$

4 Experimental Results

Many experiments have been performed by using the experimental platform as shown in Fig. 6 to illustrate the relationship between PLR (packet loss ratio) and RSSI in the WLAN or between PLR and Ec/Io in the 3G network. Fig. 7 shows the experimental results of the relationships of PLR with RSSI and Ec/Io. As shown in Fig. 7, the channel state transition of the proposed wireless channel model is performed by experimental thresholds, which are 35 of RSSI and 10.8 of Ec/Io. Using the relationship in Fig. 7, the transition probability matrix can be found to be $p0$=0.8142, $p1$=0.6657 in WLAN and $p0$=0.9545, $p1$=0.4280 in 3G network.

Using the proposed wireless channel model, we have simulated vertical handoff in different scenario to show the effectiveness of the proposed A-GBA bit allocation method. The "Foreman" sequence with 300 frames of QCIF format (176×144) is used in our experiments. The test sequence is encoded to the H.263+ CBR bitstream of 128kbps with 30fps.

The handoff process including the handoff initiation is performed from the 80^{th} frame to the 150^{th} frame when the MS moves from WLAN network to 3G network. The value of the speed factor is set to be $\alpha = (T_I + T_V) / T_I$.

As shown in Fig. 8, despite of two times vertical handoff, the 30KB client buffer reserved for buffering simulation is maintained stably in the proposed A-GBA without the underflow and the overflow caused by vertical handoff.

Fig. 9 shows the PSNR performance of the proposed A-GBA bit allocation method in vertical handoff. In Fig. 9, "non-adaptive" denotes the case that the GOP bit allocation method does not consider the latency and packet loss caused in vertical handoff. It is clearly seen that under the same conditions of vertical handoff, the

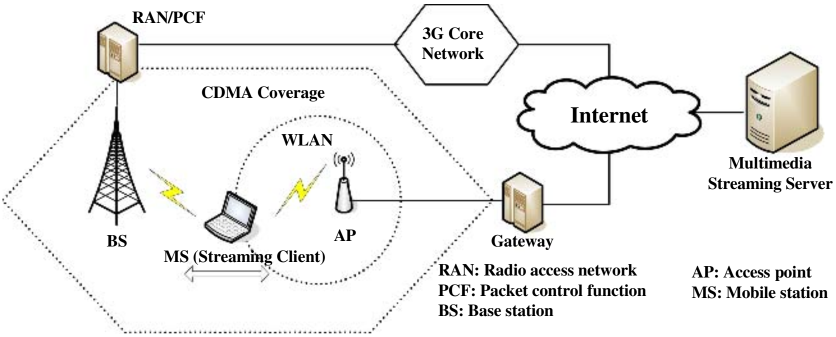
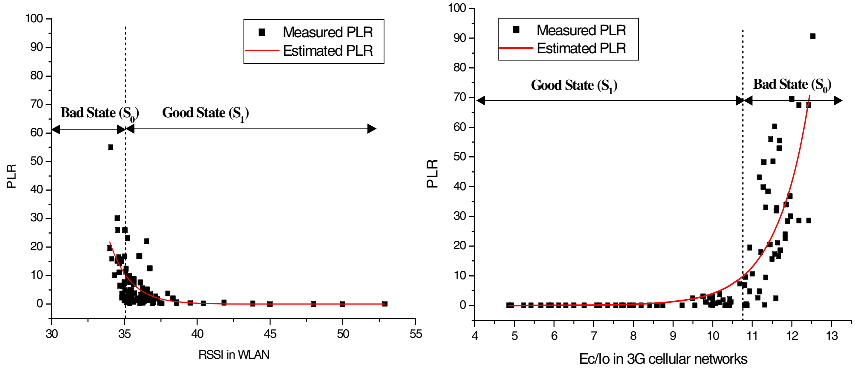


Fig. 6. Experimental platform



(a)

(b)

Fig. 7. Channel state determination: (a) PLR vs RSSI in the WLAN (b) PLR vs Ec/Io in the 3G network

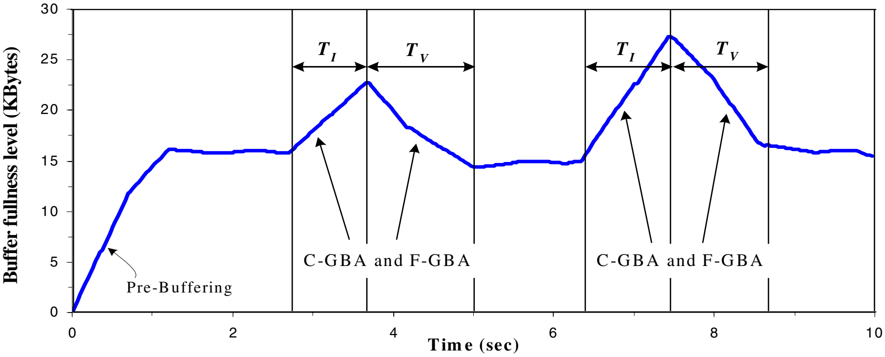


Fig. 8. Client buffer fullness level for the proposed A-GBA method

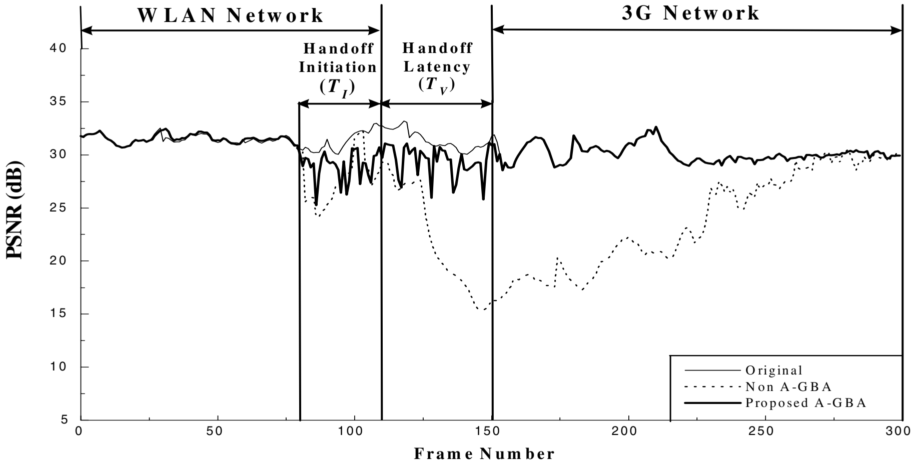


Fig. 9. PSNR performance of the proposed A-GBA method

proposed A-GBA bit allocation method can efficiently reduce the video quality degradation as compared with the non- adaptive GOP bit allocation.

5 Conclusions

The video quality degradation due to latency and packet loss caused in vertical handoff is the critical problem in seamless video streaming. To solve this problem, in this paper, we propose the A-GBA method, which includes the C-GBA and the F-GBA method. The combination of the C-GBA and F-GBA method enables the streaming server to perform the bit allocation efficiently to adapt video delivery to vertical handoff scenario. Experimental results show that the proposed method can provide significant improvements to video streaming performance, and robustness against the latency and packet losses caused by vertical handoff.

Acknowledgments. This research was supported by SAMSUNG ELECTRONICS CO., LTD. established by Network Adaptive Video Transmission Project.

References

1. McNair, J., Fang, Z.: Vertical handoffs in fourth-generation multinetwork environments. Wireless Communication. IEEE **11** (2004) 8–15
2. Wu, H., Zhang, Q., Zhu, W.: Design Study for Multimedia Transport Protocol in Heterogeneous Networks. IEEE International Conference on Communications **1** (2003) 567–571
3. Stemm, M., Katz, R. H.: Vertical handoffs in wireless overlay networks. ACM Trans. Networks and Applications **3** (1998) 335-350

4. Jung, S. K., Cho, D. H., Song, O. S.: QoS based vertical handoff method between UMTS systems and wireless LAN networks. 2004 IEEE 60th Vehicular Technology Conference **6** (2004) 4451–4455
5. Shoham, Y., Gersho, A.: Efficient bit allocation for an arbitrary set of quantizers. IEEETrans. Acoust., Speech, Signal Processing, vol. **36**, (1988) 1445-1453
6. Yuan, W., Lin, S., Zhang, Y., Luo, H.: Optimum Bit Allocation and Rate Control for H.264/AVC. IEEE Trans. Vol **16** (2006) 705 - 715
7. ISO/IED-JTC1/SC29/WG11: Test Model 5 (1993)

Memory-Efficiency and High-Speed Architectures for Forward and Inverse DCT with Multiplierless Operation

Tze-Yun Sung[1], Mao-Jen Sun[1], Yaw-Shih Shieh[1], and Hsi-Chin Hsin[2]

[1] Dept. of Microelectronics Engineering , Chung Hua University,
Hsinchu, Taiwan 30012, R.O.C.
[2] Dept. of Computer Science and Information Engineering, National Formosa University,
Hu-Wei, Taiwan 63208, R.O.C.

Abstract. Two-dimensional discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) have been widely used in many image processing systems. In this paper, efficient architectures with parallel and pipelined structures are proposed to implement 8×8 DCT and IDCT processors. In which, only one bank of SRAM (64 words) and coefficient ROM (6 words) is utilized for saving the memory space. The kernel arithmetic unit, i.e. multiplier, which is demanding in the implementation of DCT and IDCT processors, has been replaced by simple adders and shifters based on the double rotation CORDIC algorithm. The proposed architectures for 2-D DCT and IDCT processors not only simplify hardware but also reduce the power consumption with high performances.

Keywords: DCT, IDCT, parallel-pipelined architecture, memory-efficiency, high-performances.

1 Introduction

With the rapid growth of modern communication applications and computer technologies, image compression is increasingly in demand. From the compression point of view, transform coding is superior to linear prediction coding. Walsh-Hadamard transform is the simplest one, in which the computations involved in the kernel matrix are only additions and subtractions [1]. As cosine transform approximates to the optimal Karhunen-Loeve transform that is much more complicated in practice [2], the discrete cosine transform (DCT) has been widely used in the image compression task. Moreover, DCT has been adopted by the JPEG standard.

Conventionally, the double size fast Fourier transform (FFT) algorithm can be used to implement DCT. However, FFT involves complex-valued computations. Specifically, for N-point DCT, the required number of processor units is $2\log 2N$ and the order of computation time is $O(\log 2N+1)$ while applying FFT. Several fast computation algorithms were thus proposed with discussions [3]-[12], and the VLSI chip implementations of DCT for real-time applications can be found in [13]-[23].

In this paper, the CORDIC-based approach to the implementation of fast DCT and IDCT is proposed. In Section 2, the CORDIC algorithm is described briefly. In Section 3,

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 802 – 811, 2006.

both the CORDIC-based fast 2-D DCT and IDCT algorithms are presented. The implementations of the proposed low-power, parallel and pipelined architectures for 2-D DCT and IDCT are given in Section 4. Finally, conclusion can be bound in Section 5.

2 Brief Review of CORDIC Algorithm

COordinate Rotation DIgital Computer (CORDIC) is a well-known algorithm that evaluates many fundamental functions in the iterative manner [24]-[25]. As the hardware implementation of CORDIC may require only simple adders and shifters, it has received a lot of attention. A rotation of angle θ in the circular coordinate system can be obtained by performing a sequence of micro-rotations successively. Specifically, a vector can be rotated by the use of a sequence of pre-determined step-angles. The basic CORDIC algorithm in the circular coordinate system is as follows.

$$x_{i+1} = x_i - \sigma_i 2^{-i} y_i \tag{1}$$

$$y_{i+1} = y_i + \sigma_i 2^{-i} x_i \tag{2}$$

$$z_{i+1} = z_i - \sigma_i \alpha_i \tag{3}$$

where *i*=0, 1,2, …., *n*-1, and

$$\alpha_i = \text{arctan}(\ 2^{-i}\ ) \tag{4}$$

In the rotation mode, the micro-rotation direction $\sigma_i = sign(z_i)$ with $z_n \to 0$; In the vectoring mode, $\sigma_i = -sign(x_i) \cdot sign(y_i)$ with $y_n \to 0$. In the *i*-th micro-rotation, the corresponding scale factor k_i is equal to $\sqrt{1+\sigma_i^2 2^{-2i}}$. After *n* micro-rotations, the product of all the scale factors is given by

$$K_1 = \prod_{i=0}^{n-1} k_i = \prod_{i=0}^{n-1} \sqrt{1+\sigma_i^2 2^{-2i}} = \prod_{i=0}^{n-1} \sqrt{1+2^{-2i}} \tag{5}$$

One may take the iteration sequence: {0, 0, 0, 1, 2, …., *n* } for the CORDIC algorithm in the circular coordinate system to expand the convergence range of angles as follows.

$$\theta_{\max} = \text{arctan}(\ 2^{-n}\ ) + 2 \cdot \text{arctan}(\ 2^{0}\ ) + \sum_{j=0}^{n} \text{arctan}\ 2^{-j} \cong 3.3141\ (189\ ^\circ) > 180\ ^\circ \tag{6}$$

Thus, the convergence range of angles is expanded to ± 180°, and the input angle can be unlimited [26]-[27].

3 The CORDIC-Based DCT and IDCT Algorithms

The *N*-point 1-D DCT is defined as

$$Y(m) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \sqrt{2} K_m \cos\left[\frac{(2n+1)m\pi}{2N}\right] \cdot x(n) \tag{7}$$

where $m = 0,....., N-1$, $K_m = \frac{1}{\sqrt{2}}$ for $m = 0$, and $K_m = 1$ for $m > 0$.

For image applications, a separable 2-D DCT can be obtained by using the tensor product of two 1-D DCTs. Specifically, the $M \times N$ -point 2-D DCT is defined as

$$Z(u,v) = \frac{2 \cdot c(u)c(v)}{\sqrt{M \cdot N}} \cdot \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) \cdot \cos\left[\frac{(2m+1)u\pi}{2M}\right] \cdot \cos\left[\frac{(2n+1)v\pi}{2N}\right] \quad (8)$$

where $u = 0,....,M-1, v = 0,....,N-1$, $c(k) = \frac{1}{\sqrt{2}}$ for $k = 0$, and $c(k) = 1$ for $k > 0$. Equation (8) can be rewritten by

$$Z(u,v) = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} \sqrt{2} c(u) \cdot \cos\left[\frac{(2m+1)u\pi}{2M}\right] \cdot \left\{ \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \sqrt{2} c(v) \cdot \cos\left[\frac{(2n+1)v\pi}{2N}\right] \cdot x(m,n) \right\} \quad (9)$$

For 8×8 DCT, let

$$\boldsymbol{T} = \frac{1}{\sqrt{8}} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a & c & d & f & -f & -d & -c & -a \\ b & e & -e & -b & -b & -e & e & b \\ c & -f & -a & -d & d & a & f & -c \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ d & -a & f & c & -c & -f & a & -d \\ e & -b & b & -e & -e & b & -b & e \\ f & -d & c & -a & a & -c & d & -f \end{bmatrix} \quad (10)$$

where $a = \sqrt{2}\cos\left(\frac{\pi}{16}\right)$, $b = \sqrt{2}\cos\left(\frac{\pi}{8}\right)$, $c = \sqrt{2}\cos\left(\frac{3\pi}{16}\right)$, $d = \sqrt{2}\cos\left(\frac{5\pi}{16}\right)$, $e = \sqrt{2}\cos\left(\frac{3\pi}{8}\right)$, and $f = \sqrt{2}\cos\left(\frac{7\pi}{16}\right)$ .The transform coefficients $Z(u,v)$ of 8×8 DCT can be arranged and grouped into an array denoted by **Z**, which can be written by

$$\boldsymbol{Z} = \boldsymbol{T}\boldsymbol{Y}^t \quad (11)$$

where $\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{X}^t$. As a result, the separable 2-D DCT computation can be obtained by using 1-D DCT computations as follows.

$$\text{2-D DCT}(\boldsymbol{X}) = \text{1-D DCT}((\text{1-D DCT}(\boldsymbol{X}))^t \quad (12)$$

Similarly, a separable $M \times N$ -point 2-D IDCT can be obtained, which is given by

$$x(m,n) = \frac{2 \cdot c(u)c(v)}{\sqrt{M \cdot N}} \cdot \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} Z(u,v) \cdot \cos\left[\frac{(2m+1)u\pi}{2M}\right] \cdot \cos\left[\frac{(2n+1)v\pi}{2N}\right] \quad (13)$$

where $\mu = 0,....,M-1, v = 0,....,N-1$, $c(k) = \frac{1}{\sqrt{2}}$ for k=0, and $c(k) = 1$ for k>0.

Thus, the 2-D IDCT computation using 1-D IDCT computations is as follows.

$$\text{2-D IDCT}(\boldsymbol{Z})\text{=1-D IDCT}((\text{1-D IDCT}(\boldsymbol{Z}))^t) \quad (14)$$

In which, $\boldsymbol{X}=\boldsymbol{T}^t\boldsymbol{Z}\boldsymbol{T}$, $\boldsymbol{Y} = \boldsymbol{T}^t\boldsymbol{Z}^t$, and therefore

$$\boldsymbol{X}= \boldsymbol{T}^t\boldsymbol{Y}^t \tag{15}$$

3.1 Fast 1-D DCT Algorithm

Equation (10) can be further decomposed to obtain a fast algorithm for 1-D DCT [28]. Specifically, 8-point fast DCT is as follows.

$$\begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ Y(6) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ b & e & -e & -b \\ 1 & -1 & -1 & 1 \\ e & -b & b & -e \end{bmatrix} \begin{bmatrix} x(0)+x(7) \\ x(1)+x(6) \\ x(2)+x(5) \\ x(3)+x(4) \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ Y(7) \end{bmatrix} = \begin{bmatrix} a & -c & d & -f \\ c & f & -a & d \\ d & a & f & -c \\ f & d & c & a \end{bmatrix} \begin{bmatrix} x(0)-x(7) \\ -x(1)+x(6) \\ x(2)-x(5) \\ -x(3)+x(4) \end{bmatrix} \tag{17}$$

Based on equations (16) and (17), the data flow of 8-point DCT can be determined, which is shown in Figure 1. It is noted that CORDIC(2) and CORDIC(5) involved in Figure 1 have the same structure with rotation angle of $\frac{\pi}{16}$; CORDIC(3) and CORDIC(4) have the same structure with rotation angle of $\frac{5\pi}{16}$.

3.2 Fast 1-D IDCT Algorithm

By further decomposing equation (10), 8-point fast IDCT can be obtained, which is given by

$$\begin{bmatrix} x(0) \\ x(7) \\ x(4) \\ x(3) \end{bmatrix} = \begin{bmatrix} 1 & b & e & f & a & c & d \\ 1 & b & e & -f & -a & -c & -d \\ 1 & -b & -e & a & f & c & -d \\ 1 & -b & -e & -a & -f & -a & c \end{bmatrix} \begin{bmatrix} Y(0)+Y(4) \\ Y(2) \\ Y(6) \\ Y(1) \\ Y(7) \\ Y(3) \\ Y(5) \end{bmatrix} \tag{18}$$

$$\begin{bmatrix} x(1) \\ x(5) \\ x(2) \\ x(6) \end{bmatrix} = \begin{bmatrix} 1 & e & -b & c & -d & -f & -a \\ 1 & -e & -b & -d & -c & a & -f \\ 1 & -e & b & d & c & -a & f \\ 1 & e & -b & -c & d & f & a \end{bmatrix} \begin{bmatrix} Y(0)-Y(4) \\ Y(2) \\ Y(6) \\ Y(7) \\ Y(1) \\ Y(3) \\ Y(5) \end{bmatrix} \tag{19}$$

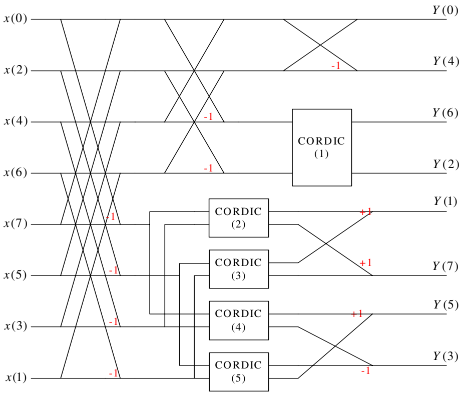


Fig. 1. Data flow of 8-point 1-D DCT

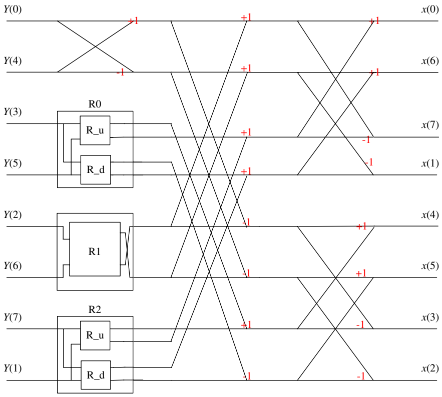


Fig. 2. Data flow of 8-point 1-D IDCT

Based on equations (18) and (19), the data flow of 8-point IDCT can be determined, which is shown in Figure 2. It is noted that the processor-R0 (with rotation angles of $\frac{5\pi}{16}$ and $\frac{\pi}{16}$) and the proceesor-R2 (with rotation angles of $\frac{\pi}{16}$ and $\frac{5\pi}{16}$) have the same structure; and the processor-R1 rotates by angle of $\frac{6\pi}{16}$.

3.3 The Proposed CORDIC-Based 2-D DCT and IDCT Architectures

Multiplication is the key operation for both DCT and IDCT. In the CORDIC-based processor with rotation mode in the circular coordinate system, multipliers of DCT and IDCT can be replaced by simple shifters and adders. Moreover, the arithmetic unit (AU) obtained by the use of double-rotation CORDIC algorithm has been taken into account to develop fast DCT and IDCT architectures. In comparison to the conventional CORDIC-based arithmetic unit, the proposed double-rotation CORDIC-based arithmetic unit can improve the latency more than 30% [29]. Thus, the hardware of arithmetic unit can be significantly saved and low-power consumption can be achieved. The overall relative error is less than 10^{-3} provided that the length of registers is 16 bits, and the number of micro-iterations in the double-rotation CORDIC processor is set to 4 [27].

4 The Proposed 2-D DCT and IDCT Processors

Based on equations (11) and (15), an efficient parallel-pipelined architecture has been developed for both 2D DCT and IDCT. Figure 3 shows the proposed architecture for 8×8 DCT and IDCT processors. In which, one SRAM bank (64 words), two 8-point DCT/IDCT processors, two multiplexers and control unit are involved. Specifically, the 8-point 1-D DCT/IDCT input-processor, which is denoted by P1, writes the intermediate result into the row and column of SRAM bank alternately. The 8-point 1-D DCT/IDCT output-processor, which is denoted by P2, reads data from the

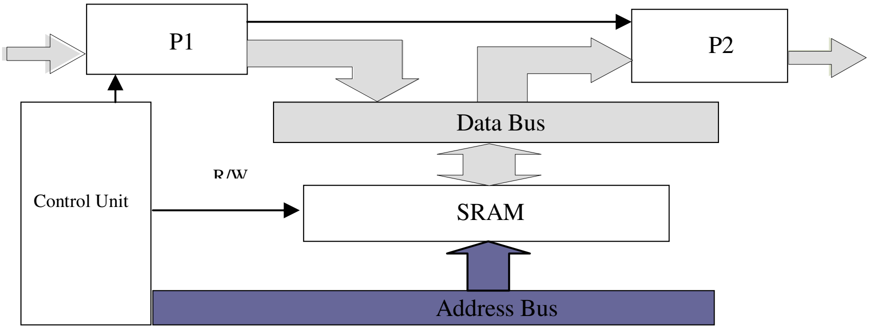


Fig. 3. The proposed architecture for 2-D DCT/IDCT processor. (P1and P2: 1-D DCT/IDCT processor).

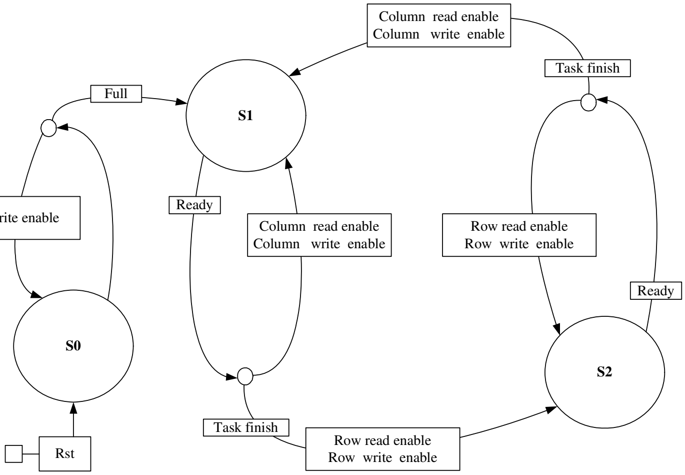
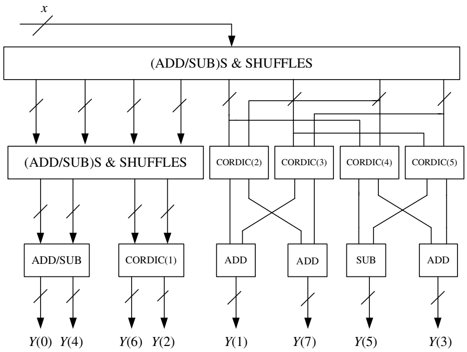


Fig. 4. The finite state machine of control unit



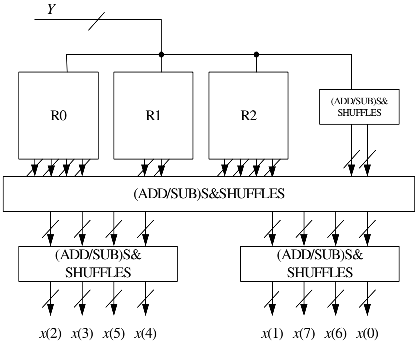


Fig. 5. The proposed 8-point DCT processor **Fig. 6.** The proposed 8-point IDCT processor

Table 1. Comparison of the proposed architecture to other commonly used architectures

8x8 2-D DCT/IDCT	Lee[9]	Chang[10]	Hsiao[11]	Hsiao[12]	Hsiao[13]	This Work
Real-adders	134	88	-	10	14	36
Real-multipliers	28	64	-	-	-	-
Rotators	-	-	-	-	3	5
Complex-multipliers	-	-	3	3	-	-
Delay elements (Words)	256	114	-	171	-	-
Memory(Words)	~384	~200	~370	-	-	70
Hardware complexity (AUs)	*O(NlogN)*	$O(N^2)$	*O(logN)*	*O(logN)*	*O(logN))*	*O(N-logN)*
Throughput (outputs/cycle)	16	8	2	2	2	8
Pipelinability	no	no	no	no	yes	yes
Parallelism	yes	yes	yes	yes	yes	yes

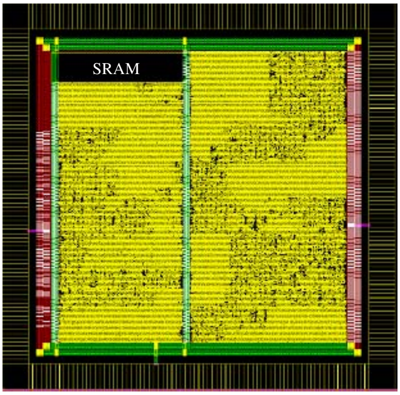


Fig. 7. The layout view of the implemented 2-D DCT processor

column and raw of SRAM bank alternately and outputs the final result. The control unit manages the data flow and arranges the timing for 2-D operations. Figure 4 shows the finite state machine (FSM) of control unit.

4.1 Implementation of the Proposed 1-D DCT and IDCT Processors

The implemented 8-point DCT/IDCT processor utilizes five CORDIC processors obtained by using the double-rotation CORDIC arithmetic [29]. Figure 5 and 6 show the proposed 8-point DCT processor and IDCT processor, respectively. As the transformation matrices involved in 1-D DCT and IDCT are column symmetry and row symmetry, respectively, the shuffle structures of DCT/IDCT processor are therefore simplified, and no multipliers are needed.

4.2 Implementation of the Proposed 2-D DCT and IDCT Processors

Figure 3 shows the proposed 2-D DCT/IDCT processor. In which, the latencies of the constituent 1-D DCT/IDCT processors are 64 clocks, hardware complexity is O(N- $\log_2 N$), and the throughputs are 8 outputs per cycle. As no multiplier is utilized in the proposed architecture, many desirable properties such as small area, low-power and high throughput are achieved. Table 1 shows the comparison of the proposed 2-D DCT/IDCT architecture to other commonly used architectures [9]-[13].

The proposed parallel-pipelined architectures for 2-D DCT and IDCT processors have been written in Verilog® and synthesized by TSMC 0.18 μm 1P6M CMOS cell libraries [30]. Their core sizes and power consumptions can be obtained from the reports of Synopsys® design analyzer and PrimPower® [31], respectively. The reported core size of 2-D DCT processor is $2372 \times 2372\,\mu m^2$, and the power dissipations of both processors are 127.7 mW at 1.8V with clock rate of 34.4MHz. Due to the limitation of paper size, the reports and layout view of the proposed 2-D IDWT processor are not presented here. Figure 7 shows the layout view of the implemented 2-D DCT processor. The proposed 2-D DCT and IDCT processors are much suited to the applications of both JPEG and MPEG standards.

5 Conclusion

By taking into account the symmetry properties of the fast DCT/IDCT algorithm, high efficiency architectures with a parallel-pipelined structure have been proposed to implement DCT and IDCT processors. For image applications, a separable 2-D DCT/IDCT can be obtained by using the tensor product of two 1-D DCT/IDCT operations. Thus, the proposed 2-D DCT/IDCT processor is composed of two successive 1-D DCT/IDCT kernels. In the constituent 1-D DCT/IDCT processors, the double-rotation CORDIC algorithm with rotation mode in the circular coordinate system has been utilized for the arithmetic unit (AU) of both DCT and IDCT, i.e. the multiplication computation. The proposed DCT/IDCT architectures are not only regularly structured but also highly scalable and flexible as well. Thus, they are much suited to VLSI implementation with design trade-offs.

References

1. Elliott, D. F., Kao K. R., “Fast Transforms Algorithms, Analysis, Applications,” Chapter 8, Walsh-Hadamard Transform, Prentice-Hall, (1982), 301-303.
2. Clarke, R. J., “Relation between the Karhenen Loeve and Cosine Transform,” IEEE Proceedings, Part F, Vol. 128, No. 6, (1981), 359-360.
3. Narasimha, M. J., Peterson, A. M., “On the Computation of the Discrete Cosine Transform,” IEEE Transactions on Communications, Vol. 26, No. 6, June 1978, pp. 934-936.
4. Haralick, R. M. “A Storage Way to Implement the Discrete Cosine Transform,” IEEE Transactions on Computers, (1976), 764-765.
5. Chen, W. H., Smith, C. H., Fralick, S. C., “Fast Computational Algorithm for the Discrete Cosine Transform,” IEEE Transactions on Communications, Vol. 25, No. 9, (1977), 1004-1009.

6. Sung, T. Y., “VLSI Parallel and Distributed Computation Algorithms for DCT Processors,” Proceedings IEEE International Phoenix Conference on Computer and Communications, Scottsdale, Arizona, USA, (1990), 121-125.
7. Sung, T. Y., “VLSI Parallel and Distributed Processing Algorithms for Multidimensional Discrete Cosine Transforms,” 1990 A Two-Track International Conference on Databases, Parallel Architectures, and their Applications, Miami Beach, Florida, USA, (1990), 36-39.
8. Sung, T. Y., “Novel Parallel VLSI Architectures for Discrete Cosine Transforms,” Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, Albuquerque, New Mexico, USA, (1990), 998-1001.
9. Lee, Y. P., Chen, T. H., Chen, L. G., Ku, C. W.,“ A Cost-Effective Architecture for 8×8 two-dimensional DCT/IDCT Using Direct Method,” IEEE Transactions on Circuits Systems for Video Technology, Vol. 7, No. 1, (1997), 459-467.
10. Chang, Y. T., Wang, C. L., “New Systolic Array Implementation of the 2-D Discrete Cosine Transform and Its Inverse,” IEEE Transactions on Circuits Systems for Video Technology, Vol. 5, No. 1, (1995), 150-157.
11. Hsiao, S. F., Shiue, W. R., “A New Hardware-Efficient Algorithm and Architecture for Computation of 2-D DCTs on a Linear Array,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, (2001), 1149-1159.
12. Hsiao, S. F., Tseng, J. M., “New Matrix Formulation for Two-Dimensional DCT/IDCT Computation and its Distributed-Memory VLSI Implementation,” IEE Proc.-Vis. Image Signal Process, Vol. 149, No. 2, (2002), 97-107.
13. Hsiao, S. F. , Hu, Y. H., Juang, T. B., Lee, C. H., “Efficient VLSI Implementations of Fast Multiplierless Approximated DCT Using Parameterized Hardware Modules for Silicon Intellectual Property Design,” IEEE Trans. Circuits and Systems, Part-I: Regular Papers, Vol. 52, No. 8, (2005), 1568-1579.
14. Srinvasan, V., Liu, K. J. R., “VLSI Design of High-Speed Time-Recursive 2-D DCT/IDCT Processor for Video Applications,” IEEE Transactions on Circuits Systems for Video Technology, Vol. 6, No. 1, (1996), 87-96.
15. Kuroda, T., “A 0.9-V, 150-MHz, 10-mW, 4mm2, 2-D Discrete Cosine Transform Core Processor with Variable Threshold-Voltage(VT) Scheme,” IEEE Journal of Solid-States Circuits, Vol. 31, No. 11, (1996), 1770-1778.
16. Rambaldi, R., Uguzzoni, A., Guerrieri, R., “A 35 μ W 1.1 V Gate Array 8×8 IDCT Processor for Video-Telephony,” Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, (1998), 2993-2996.
17. Chen, T. H., “A Cost-Effective 8×8 2-D IDCT Core Processor with Folded Architecture,” IEEE Transactions on Consumer Electronics, Vol. 45, No.2, (1999), 333-339.
18. Sung, T. Y., Sung, Y. H., “A Novel Implementation of Cost-Effective Parallel- Pipelined 8×8 DCT Processor,” The Fourth IEEE Asia-Pacific Conference on Advanced System Integrated Circuits (AP-ASIC) 2004, Fukuoka, Japan, (2004), 200-203.
19. Hu, Y. H., Wu, Z., “An Efficient CORDIC Array Structure for the Implementation of Discrete Cosine Transform”, IEEE Transactions on Signal Processing, Vol. 43, No. 1, (1995), 331-.336.
20. Jeong, H., Kim, J, Cho, W. K., “Low-Power Multiplierless DCT Architecture Using Image Data Correlation,” IEEE Transactions on Consumer Electronics, Vol. 50, No. 1, (2004), 262-267.
21. Gong, D., He, Y, Gao, Z., “New Cost-Effective VLSI Implementation of a 2-D Discrete Cosine Transform and Its Inverse”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 4, (2004), 405-415.

22. Dimitrov, V., Wahid, K., Jullien, G., “Multiplication-Free 8×8 2D DCT Architecture Using Algebraic Integer Encoding”, Electronics Letters, Vol. 40, No. 20, (2004).
23. Alam, M., Badawy, W., Jullien, G., “A New Time Distributed DCT Architecture for MPEG-4 Hardware Reference Model”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 5, (2005), 726-730.
24. Volder, J. E., “The CORDIC Trigonometric Computing Technique,” IRE Transactions on Electronic Computers, Vol. EC-8, (1959), 330-334.
25. Walther, J. S., “A Unified Algorithm for Elementary Functions,” Spring Joint Computer Conference Proceedings, Vol.38, (1971), 379-385.
26. Hu, X., Harber, R. G., Bass, S. C., “Expanding the range of the Convergence of the CORDIC Algorithm”, IEEE Transactions on Computers, Vol. 40, No. 1, (1991), 13-21.
27. Sung, T. Y., Sung, Y. H., “The Quantization Effects of CORDIC Arithmetic for Digital Signal Processing Applications”, The 21st Workshop on Combinatorial Mathematics and Computation Theory, Taiwan, (2004), 16-25.
28. Sung, T. Y., “A Memory-Efficient and High-Speed Split-Radix FFT/IFFT Processor Based on Pipelined CORDIC Rotations,” to appear in IEE Proceedings – Vision, Image and Signal Processing, (2006).
29. Sung, T. Y., Chen, C. S., Shih, M. C., “The Double Rotation CORDIC Algorithm: New Results for VLSI Implementation of Fast Sine/Cosine Generation,” 2004 International Computer Symposium (ICS-2004), Taipei, Taiwan, (2004), 1285-1290.
30. 30.“TSMC 0.18 μm CMOS Design Libraries and Technical Data, v.3.2,” Taiwan Semiconductor Manufacturing Company, Hsinchu, Taiwan, and National Chip Implementation Center (CIC), National Applied Research Labs., Hsinchu, Taiwan, R.O.C., (2006).
31. Synopsys FPGA Express, http://www. synopsys.com/ products.

Unequal Priority Arrangement for Delivering Streaming Videos over Differentiated Service Networks

Chu-Chuan Lee[1], Pao-Chi Chang[2], and Shih-Jung Chuang[2]

[1] Chunghwa Telecommunication Laboratories, R.O.C.
[2] Department of Communication Engineering, National Central University, R.O.C.
{cclee, pcchang, sjchuang}@vaplab.ee.ncu.edu.tw

Abstract. Video packets have different significances due to the video coding property. When delivering video data without priority strategy over the Internet, it will seriously degrade the received picture quality. This paper proposes an Unequal Priority Arrangement (UPA) mechanism for video transmission on differentiated service networks. UPA determines the priority of a video packet according to the evaluation from both temporal and spatial domains simultaneously. From the temporal domain, UPA evaluates the packet significance based on the temporal position of the packet and the induced error propagation if the packet is lost. From the spatial domain, UPA evaluates the packet importance based on its content, where the ratio of intra-refreshed MBs is used. Moreover, according to the video complexity, UPA can flexibly adjust the weight based on the evaluation results from temporal and spatial domains. Simulation results show that delivering video data with UPA on differentiated service network outperforms traditional temporal-based and spatial-based priority strategies up to 1.8 dB and 1 dB, respectively.

Keywords: Differentiated services, streaming videos, unequal priority arrangement.

1 Introduction

Most video coding methods exploit both temporal and spatial redundancies to reduce required transmission rate and achieve high compression efficiency. In the spatial domain, there exists a high correlation within a picture. In the temporal domain, there usually exists a high similarity between successive pictures. However, the received video quality is highly sensitive to packet loss. When a video packet that belongs to I-frame is lost due to network congestion, all frames belonging to the same GOP (Group of Picture) are hurt due to error propagation in the decoding process. This phenomenon causes significant degradation of received picture quality. Moreover, all succeeding frames belonging to the same GOP are also hurt if a video packet that belongs to P-frame is lost, as shown in Fig. 1. Therefore, a robust network mechanism that can provide sufficient protections to video data is essential for received picture quality.

Unfortunately, the default QoS (Quality of Service) strategy of Internet is the best-effort transmission, which is lack of QoS guarantee to encoded video data. In order to enhance the QoS capability of Internet, the Internet Engineering Task Force (IETF)

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 812 – 821, 2006.

had proposed two QoS approaches: Integrated Services (InteServ) [1] and Differentiated Services (DiffServ) [2]. InteServ can provide the flow-based QoS guarantee to delivered data. However, InteServ has the scalability problem due to its high implementation complexity. DiffServ was thus proposed to solve the above scalability problem and was popularly implemented in network equipments. In DiffServ, each packet is assigned and classified to one of few classes. However, if much video data with the same class arrive to a network equipment simultaneously, DiffServ cannot provide the absolute delay and loss guarantees to each video flow. That is, the congestion loss and delay tolerance violation of video packets are still possible to happen even the DiffServ network is implemented. DiffServ can only provide the class-based QoS guarantee to delivered data, which is not sufficient to requirements of compressed video traffic.

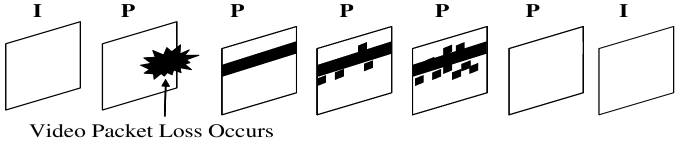


Fig. 1. Error propagation effect if a video packet of P-frame is lost

To prevent the unexpected packet loss of significant video frames such as I-frames in DiffServ network, an unequal priority assignment scheme is required for video packets at the video sender side. The priority of video packet also implies the distortion effect induced by packet loss. Many research results were developed in past few years. In [3] and [4], the frame type (I-/P-/B-frame) is directly used to classify the priorities of video packets. Video packets that belong to I-frame and B-frame always have the highest and the lowest priorities, respectively. However, the error propagation effect of each packet in the temporal domain is not considered. In [5], the error propagation influence of each packet is estimated at the sender side according to its temporal position in a GOP. Two packets that belong to different P-frames have different priorities. However, the method in [5] still ignored the content diversity among these video packets that belong to the same video frame. In [6], the technique of intra-refreshed MacroBlock (MB) is used to alleviate error propagation. The content of each packet is evaluated for determining the packet significance in the spatial domain. According to the ratio of the number of intra-refreshed MBs to the total number of MBs in a packet, the method of [6] defined the Quality Enhancement (QE) of a given video packet. However, the error propagation effect of each packet in the temporal domain is not examined in [6].

This study proposes an Unequal Priority Arrangement (UPA) mechanism for video transmission on differentiated service networks. In contrast to traditional temporal-based or spatial-based priority assignment methods [5][6], UPA determines the priority of a video packet according to the evaluation from both temporal and spatial domains simultaneously. From the temporal domain, UPA evaluates the packet significance based on the temporal position of the packet and the induced error propagation if the packet is lost. From the spatial domain, UPA evaluates the packet importance based on its content, where the ratio of intra-refreshed MBs is used. Moreover, according to the video complexity, UPA can flexibly adjust the weight based on the evaluation results from temporal and spatial domains. Regarding a video sequence with low motion variation and low complexity, the evaluation from spatial domain dominates the packet priority. In contrast, the evaluation from temporal domain dominates the packet priority if a video sequence has high motion variation and high complexity.

The remaining of this paper is organized as follows. The detailed process of UPA is presented in Section 2. Simulation environment and results are discussed in Section 3. Finally, Section 4 concludes this paper.

2 Unequal Priority Arrangement Strategy

2.1 Determining Significance Grade of Video Packets

Since a lost video packet affects the received picture quality from the temporal and spatial domain simultaneously, UPA integrates and improves the methods utilized in [5] and [6]. The *i*-th packet of the *k*-th video frame in a GOP is denoted as VP_{ki}. This study then estimates the Significance Grade (SG) of VP_{ki} as

$$\begin{aligned} SG_{ki} &= (1-\alpha)\cdot SGT_{ki} + \alpha \cdot SGS_{ki} \\ &= (1-\alpha)\left(\sqrt{(N_I - k+1)\cdot Q_{ki}}\right) + \alpha\left(\frac{MB_{Total\_ki} + MB_{Intra\_\_ki}}{MB_{Total\_ki}}\right) \end{aligned} \tag{1}$$

where SGT_{ki} and SGS_{ki} represent the estimated significance grades of VP_{ki} from the temporal and spatial domain, respectively. Regarding the SGT_{ki}, this study assumes that the cumulative error propagation of succeeding video frames of VP_{ki} by a geometric progression with common ratio *r*=1. From [7] we find that the error propagation effect strongly depends on the frame position in the GOP, while it is almost independent of the sequence. On the other hand, SGS_{ki} calculates the ratio of the number of intra-refreshed MBs in VP_{ki} to the number of total MBs in VP_{ki}. In this paper, a periodically intra-refresh operation is executed.

In Eq. (1), α is a weighting factor that determines the contributions of SGT_{ki} and SGS_{ki} to SG_{ki}. The value of α depends on the complexity of video sequence, which will be discussed in Section 2-2. N_I represents the length of a GOP, MB_{Total_ki} represents the number of MBs and MB_{Intra_ki} represents the number of intra-MBs in VP_{ki}. The error concealment effect while VP_{ki} is lost is expressed by

$$Q_{ki} = \frac{\sum_j |PO(j) - PE(j)|^2}{\sum_j |PO(j)|^2} \tag{2}$$

where *j* is the pixel index, $PO(j)$ is the original encoded value of pixel *j*, and $PE(j)$ is the recovered value of pixel *j* when VP_{ki} is lost and error concealment is activated. The detailed procedure of UPA is shown in Fig. 2.

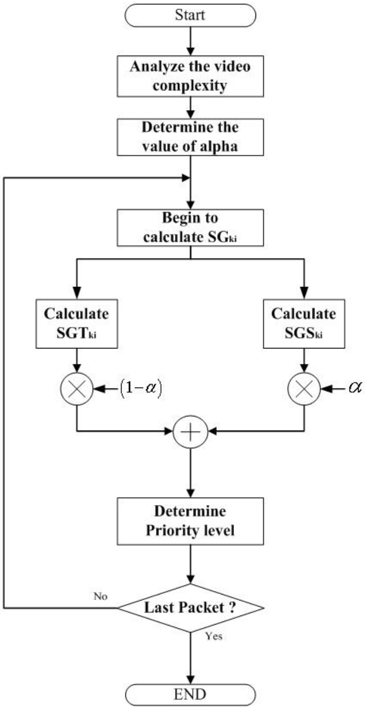


Fig. 2. Detailed operations of UPA

Figures 3 to 5 present the results of executing UPA for "*Foreman*" sequence, where α is set to 1, 0, 0.25, respectively. Figs. 3 and 4 show two extreme cases, that is, Spatial Domain Consideration Only (SDCO) case and Temporal Domain Consideration Only (TDCO) case. Considering the SDCO case that is similar to [6], the height of a vertical line represents the capability of a packet to enhance the received picture quality, as shown in Fig. 3. If the height of a vertical line is explicitly higher than that of adjacent vertical lines, the packet that corresponds to the higher vertical line contains more intra-refreshed MBs and thus provides more contribution to the received quality than other packets. Note that the consideration of error propagation effect is not presented in Fig.3. Regarding the TDCO case that is similar to [5], the height of vertical line gradually decreases when the packet number increases, as shown in Fig. 4. This is mainly because the error propagation resulted from the first erroneous frames of a GOP is larger than that from the later erroneous frames in the same GOP. Note that the content significance of packet cannot be observed in Fig.4.

Regarding the normal case of UPA that is proposed by this paper, the height of a vertical line consists of SGT_{ki} and SGS_{ki}, as shown in Fig. 5. The significance grade of a packet does not only depend on the packet content, but also depend on the temporal position of the packet in a GOP. There present two examples in Fig. 5. In the first example, the significance of Packet B is higher than that of Packet A, even the

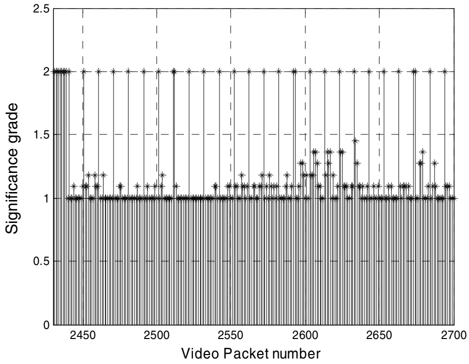


Fig. 3. Results of UPA, α=1

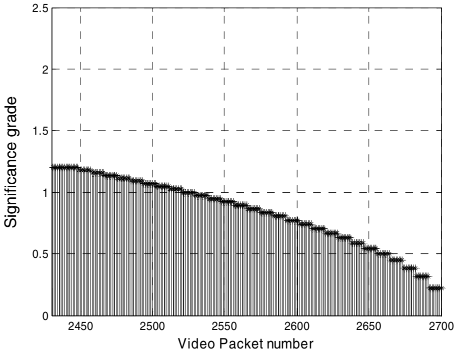


Fig. 4. Results of UPA, α=0

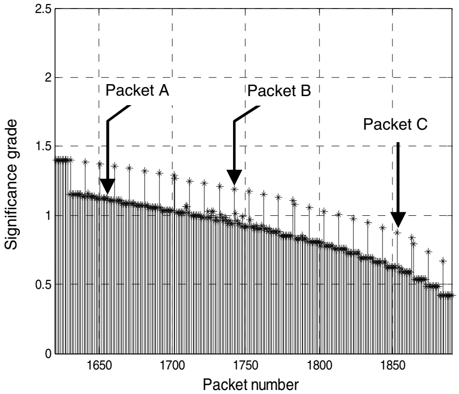


Fig. 5. Results of UPA, α=0.25

temporal position of Packet B lag behind Packet A. In the second example, the significance of Packet C is lower than that of Packet A, even the value of SGS_{ki} of Packet C is larger than that of Packet A. The capability that estimates the significance of packets from both the temporal and spatial domains is the major contribution of this study, which can provide more accurate judgment of packet importance than other traditional methods.

2.2 Determining Weighting Factor α

Considering a video sequence with low motion variation and low complexity, the similarity between adjacent video frames is high and the error propagation effect in the temporal domain is thus low. Therefore, it is intuitive that the contribution of SGT_{ki} to SG_{ki} should be lower than that of SGS_{ki}. Based on the similar concept, UPA flexibly adjusts the weighting factor α between SGT_{ki} and SGS_{ki} according to the video complexity. Referring to common classification of video sequence, this study defines that Class A video sequences have low motion variation and low complexity such as Akyio, Class B video sequences have high motion variation and low complexity such as Foreman, and Class C video sequences have high motion variation and high complexity such as Stefan.

Figure 6 shows error propagation results of video sequences that belong to different classes, where some video packets of the first *N* frames in a GOP are assumed to be lost. Observing on Fig. 6, the error propagation of *Stefan* is serious because of its high motion variation and high complexity. Therefore, the temporal position of the lost packet is very important to the error propagation effect of Class C video sequences. In contrast, considering the "*AKyio*", the error propagation effect is not obvious. Therefore, the weight of SGT_{ki} should be lower than that of SGS_{ki} in (1) while considering Class A video sequences. This paper summarizes the above observations and determines various ranges of α for different video classes, as presented in Table 1.

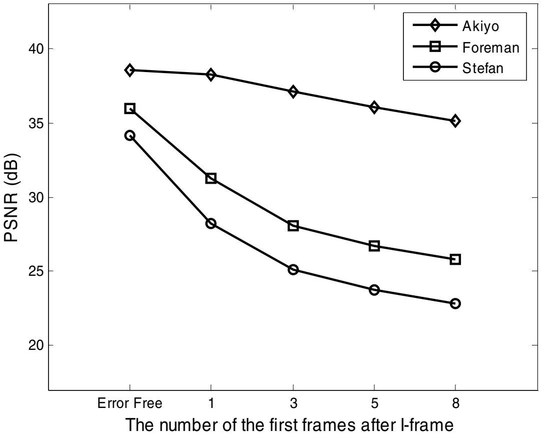


Fig. 6. Error propagation effect of different videos

Table 1. Range of α for different video classes

Video Class	Value of α
Class A	0.9 ~ 0.6
Class B	0.6 ~ 0.4
Class C	0.4 ~ 0.1

3 Simulation Results and Discussions

In this section, the performance of UPA is evaluated by simulations. This paper uses Network Simulator version 2 (NS-2) to simulate the Diffserv network. In these simulation cases, this study uses the H.264 codec and compresses videos at a target rate of 750K bps. The video format is CIF and the frame rate is 30 frames per second. In addition, the length of GOP is set to 30 frames and the cyclic intra refreshment function is activated. As shown in Fig.7, video flows have to compete with background flows and the bottleneck link varies its bandwidth dynamically. Three differentiated service levels are provided in the simulated network, where the high priority level, the medium priority level, and the low priority level are named as P1, P2, and P3, respectively. A Weighted Round Robin (WRR) is also utilized here and the Drop Tail operation is activated for queue management.

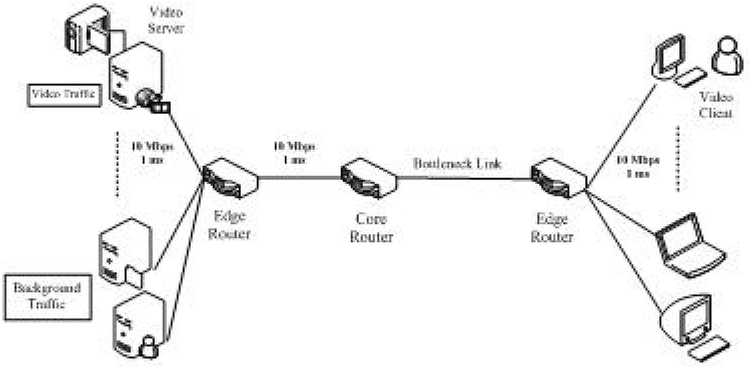


Fig. 7. Simulation architecture using NS-2

Referring to (1), the estimated significance grade of packet is widely distributed from 0 to 2. Therefore, the mapping between the value of significance grade and the limited differentiated service levels is required, which is called QoS mapping [8]. This study defines three QoS mapping scenarios for different Available Bandwidth (AB) conditions, as shown in Fig. 8. A TCP Friendly mechanism integrated with feedback function is activated in these simulation cases. When the network available bandwidth changes, the video server selects a corresponding QoS mapping scenario based on rules presented in Fig. 8.

Figure 9 shows performance comparisons of UPA with TDCO and SDCO methods, where the "*Foreman*" sequence is used. The TDCO scheme with α = 0 considers the significance of packet from the temporal domain only and the SDCO method with α = 1 considers the significance of packet from the spatial domain only. Considering the normal case of UPA, the value of α is set to 0.5. From Fig. 9, it is obvious that the received PSNR using UPA is better than that using TDCO or SDCO. This is mainly because UPA does not only refer to the packet content but also refer to the temporal position of the packet in a GOP. Using UPA, the improvement to received PSNR is significant when the packet loss rate increases. In addition, the received picture quality degraded significantly if the network only provides best-effort delivery quality to video data. Figure 10 shows the results while delivering the "*Stefan*" sequence. Note that the value of α in the normal case of UPA is changed to 0.7 since the "*Stefan*" sequence is classified as Class C Video.

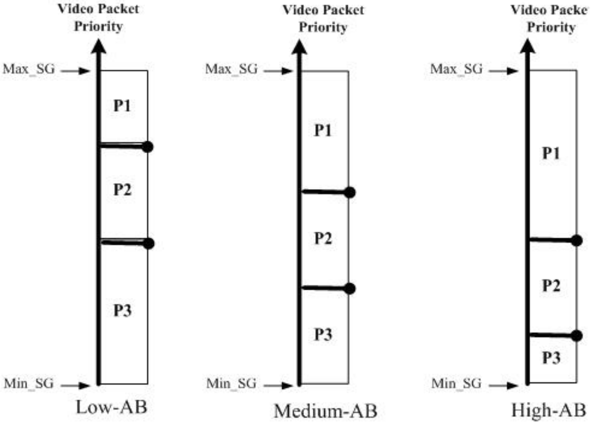


Fig. 8. QoS mapping between SG and differentiated service levels

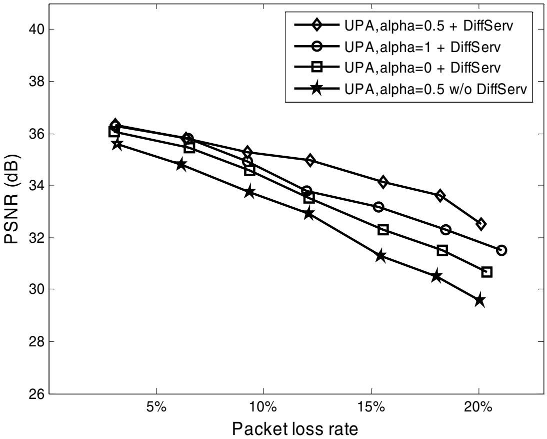


Fig. 9. Performance comparison of UPA with TDCO and SDCO, where *Foreman* is used

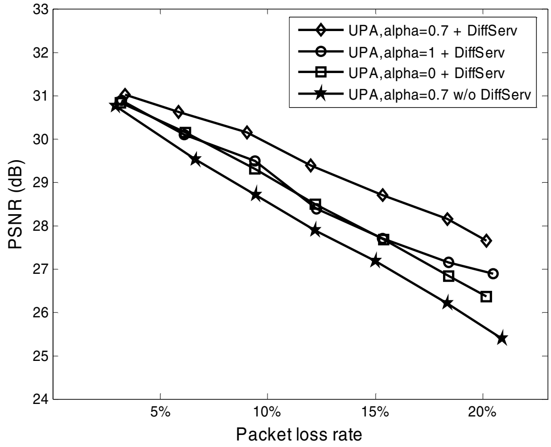


Fig. 10. Performance comparison of UPA with TDCO and SDCO, where *Stefan* is used

Finally, this paper has tried to include more characteristics of H.264, such as the data partition, into the consideration of UPA. However, we found that the average payload length of video packet decreases significantly, after classifying the significance of video data by many characteristics of H.264. This phenomenon significantly increases the packet header overhead and reduces the overall delivery performance. From the packetization viewpoint, the tradeoff between the error resilience functions of H.264 and the introduced header overhead is an open issue for further study.

4 Conclusions and Future Works

When estimating the significance of video packets, the evaluation from the temporal domain only or the spatial domain only is not sufficient. The UPA mechanism proposed by this paper does not only refer to the packet content from the spatial domain, but also refer to the temporal position of the packet in a GOP from the temporal domain. Simulation results show that delivering video data with UPA on differentiated service network outperforms traditional temporal-based-only and spatial-based-only priority strategies up to 1.8 dB and 1 dB, respectively. In the future, this study will target at the optimization works to the determination of α based on the video characteristics and to the QoS mapping between the significance grade and the limited differentiated service levels in network.

Acknowledgments. The authors would like to thank C. H. Chuang for his precious assistances during the simulation process and his valuable comments for this study.

References

1. Braden.R., Clark.D., Shenker.S.:Integrated Services in the Internet Architecture: an Overview. RFC 1633, June (1994).
2. Blake.S., Black.D., Davies.E.:An Architecture for Differentiated Services. RFC2475, Dec. (1998).

3. Magalhaes.J., Guardieiro.P.: A New QoS. Mapping for Streamed MPEG Video over a DiffServ Domain. Communications, Circuits and Systems and West. Sino Expositions, IEEE 2002 International Conference on Volume 1, July (2002) 675-679.
4. C.-H.Ke., C.-K.Shieh., W.-S.Hwang, Ziviani.A.:A Two-Markers System for Improved MPEG Video Delivery in a DiffServ Network. IEEE Communications Letters, IEEE Press, v. 9, April (2005) 381-383.
5. Zhang.F., Pickering.M.R., Frater.M.R., and Arnold.J.F.: Optimal QoS mapping for streaming video over Differentiated Services networks. Acoustics, Speech, and Signal Processing, 2003. Proceedings.2003 IEEE International Conference on Volume 5 (2003).
6. Cote.G., Kossentini.F.:Optimal Intra Coding of Blocks for Robust Video Communication over The Internet. Signal Processing: Image Communication, Sep. (1999) 25-34.
7. Fabio De Vito, Davide Q., Juan Carlos De Martin: Model-based distortion estimation for perceptual classification of video packets. IEEE 6th Workshop on Multimedia Signal Processing (2004).
8. Shin.J., Kim.J., C.-C.J.Kuo:Quality-of-Service Mapping Mechanism for Packet Video in Differentiated Services Network. IEEE Transactions on Multimedia, Vol. 3, No. 2, June (2003) 219-231.

A New Macroblock-Layer Rate Control for H.264/AVC Using Quadratic R-D Model

Nam-Rye Son, Yoon-Jeong Shin, Jae-Myung Yoo, and Guee-Sang Lee

Dept. Computer Science, Chonnam National University, Youngbong-dong Pug-gu,
Gwang-ju, South Korea
{nrson, gslee}@chonnam.ac.kr, syj@gwangju.ac.kr,
jmyoo@cad.chonnam.ac.kr

Abstract. In recent years, rate control is an important technique in real time video communication applications using H.264/AVC. Many existing rate control algorithms employ the quadratic rate-distortion (R-D) model, which is determine the target bits for each I, P, B frame. In this paper, we analysis the problems in rate control for JVT video coding using quadratic R-D model. According to the analysis and experimental results, we present a new frame-layer rate control scheme for JVT video coding based on the quadratic R-D model. Also we estimate the target bit rate for macroblock-layer effectively. The experimental results show that with many scene changes between neighboring frames, existing algorithm exceeded transmission bit rate, while proposed algorithm tranquilized bit rate for stable delivery.

Keywords: H.264 video coding, Quadratic R-D Model, Mean Absolute Difference.

1 Introduction

H.264/AVC is the newest international video coding standard. By the time of this publication, it is expected to have been approved by ITU-T as recommendation H.264 and by ISO/IEC as International Standard 14496-10 (MPEG-4 part 10) Advanced Video Coding(AVC)[1]. The H.264/MPEG-4 AVC video compression standard promises a significant improvement over all previous video compression standards. In terms of coding efficiency, the new standard is expected to provide at least 2x compression improvement over the best previous standards and substantial perceptual quality improvements over both MPEG-2[2] and MPEG-4. The standard, being jointly developed by ITU-T and ISO/IEC, will address the full range of video applications including low bit rate wireless applications, SD(Standard Definition) and HD(High Definition) broadcast television, video streaming over the Internet, delivery of HD-DVD content, and the highest quality video for digital cinema applications.

One important component of video codec is rate control. It is a necessary part of an encoder to allocate the suitable number of bits to each video frame and then smooth out the variable bit rate to constant bit rate channel[3]. Rate control algorithms have been widely studied in standards, like MPEG-2, MPEG-4, H.263. For H.264/AVC, there are two different rate control algorithms proposed in the literature. JVT-D030 is

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 822 – 831, 2006.

based on MPEG-2 TM5 rate control[4] but is not suitable for low bit rate video applications. Our study bases on another scheme, which is based on fluid traffic model, linear model, and Rate Distortion Optimization (RDO)[5].

In short, since existing bit rate control scheme for JVT video coding codes current macroblock without accurately acknowledge of MAD(Mean Absolute Difference) to be coded. Because this, it cannot deal with drastically change of delivery channel status or video characteristics. Accordingly a new bit rate control scheme for H.264/AVC codec is needed considering characteristics of inter-frames.

The rest of the paper is organized as follows. In Section 2, we address the existing JVT-H014 video coding[5] and present the problem in the quadratic R-D model and in the rate control for JVT video coding. Then in Section 3, we propose a new macroblock-layer rate control using improved MAD. The experiment and results are presented in the section 4. Finally concluding remarks are given in Section 5.

2 Problems in Rate Control for JVT Video Coding

2.1 Estimated Bit Rate for GOP and Allocated the Target Bit Rate

According to [5], frame layer rate control scheme consists of two stages: pre-encoding and post-encoding. The objective of the first stage is to compute Quantization Parameter (QP) for all frames and it composed of two sub steps: (a) determine a target bit rate for each P frame and (b) compute the QP and perform RDO. In the pre-encoding stage, a quadratic R-D model is used to calculate the corresponding QPs, in the post-encoding stages, the model parameters are continually updated and buffer's control is performed. In this section, we summarize and analysis the JVT-H014 video coding used for estimating the target bits.

To estimate target bits for the current frame, a fluid traffic model is employed, which is based on the linear tracking theory[9]. For simplicity, let us assume one Group of Pictures (GOP) is used and the video sequence is encoded first as an I frame, and subsequently P frames. To illustrate rate control modeling, Let N denote the total number of frames in a GOP, $n_j(j=1,2,...N)$ denotes the j^{th} frame, and $B_c(n_j)$ denotes the occupancy of virtual buffer after coding the j^{th} frame. The fluid traffic model is stated as

$$\begin{aligned} B_c(n_1) &= \frac{B_s}{8} \\ B_c(n_{j+1}) &= \min\{\max\{0, B_c(n_j) + A(n_j) - \frac{u(n_j)}{F_r}\}, B_s\} \end{aligned} \tag{1}$$

Where B_s is the buffer size, $A(n_j)$ is the number of bits generated by the j^{th} frame, $u(n_j)$ is the available channel bandwidth, and F_r is the predefined frame rate. The determination of target bits for current P frame is composed of two steps.

Step 1. Determination of target buffer occupancy

Since the QP of the first P frame is given at the GOP layer in this algorithm, the initial value of target buffer level is set as

$$Tbl(n_2) = B_C(n_2) \tag{2}$$

Then the target buffer levels of other P frames in the GOP are predefined by using the following function

$$Tbl(n_{j+1}) = Tbl(n_j) - \frac{Tbl(n_{j-1}) - B_s / 8}{N_p - 1} \tag{3}$$

Where N_p is the total number of P frames in the GOP.

Step 2. Computation of target bit rate

By using linear tracking, the target bit allocated for the j^{th} frame is determined based on the target buffer level, the frame rate, the available channel bandwidth, and the actual buffer occupancy as follows:

$$T_{buf} = \frac{u(n_j)}{F_r} + \gamma(Tbl(n_j) - B_c(n_j)) \tag{4}$$

Where γ is a constant and its typical value is 0.75. Meanwhile, the remaining bits are also computed as like function

$$T_r = \frac{R_r}{N_r} \tag{5}$$

Where R_r is the number of bits remaining for encoding this sequence, and N_r is the number of P frames remaining for encoding. The final target bit T is a weighted combination of T_r and T_{buf}.

$$T = \beta \times T_r + (1-\beta) \times T_{buf} \tag{6}$$

Where β is a weighting factor and its typical value is 0.5.

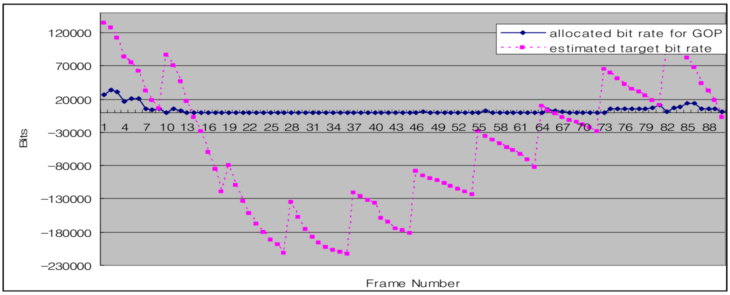


Fig. 1. Estimated target bits and allocated bit rate of GOP for TABLE TENNIS sequence

After the encoding results for TABLE TENNIS sequence, Figure 1 shows allocated bit rate for GOP and estimated target bit rate using JVT-H014 algorithm. In detail analysis, the value of target bit rate is '0' or 'positive' value because of considering both the status of buffer occupation after encoding and remained bit rate in the *eq*.(1). Specially, in the case of scene change, the value of allocated bit rate for GOP is 'negative' because estimated target bit rate is predicted wrongly, namely estimated target bit rate is greater than estimated buffer occupation.

2.2 Rate Control for JVT Video Coding Using Quadratic Rate-Distortion Model

This quadratic R-D model[6] is utilized for QP determination in rate control scheme, where the quadratic R-D model's parameters are estimated using MAD, which is prediction error between previous and current frame. Since quantization parameters are specified in both rate control and RDO, there exists a problem when the rate control is implemented: to perform RDO for MB, a quantization parameter should be first determined for the MB by using the MAD of MB. We take the following specific form of the simple MAD as like figure 2.

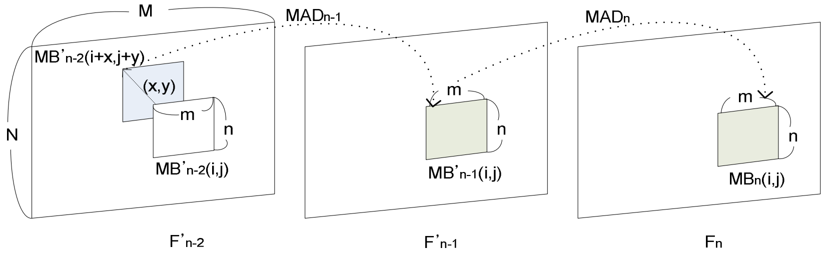


Fig. 2. Prediction of MAD for JVT video coding

$$
\begin{aligned}
MAD_{n-1}(i,j) &= \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left| MB_{n-1}(i,j) - MB_{n-2}(i+x, j+y) \right| \\
MAD_n &= a_1 \times \frac{1}{X \times Y} \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} MAD_{n-1}(i,j) + a_2, \left(X = \frac{M}{m}, Y = \frac{N}{n}\right)
\end{aligned}
\tag{7}
$$

In *eq*. (7), MAD_n is the mean absolute difference between pre-previous frame(F'_{n-2}) and previous frame(F'_{n-1}) in the JVT video coding, when current frame(F_n) is encoding. And MAD_{n-1} is predicted MAD of previously encoded P frame $(n-1)^{th}$. Where a_1 and a_2 are the two coefficients of the prediction model. The initial values of a_1 and a_2 are to 1 and 0, respectively. They are updated after coding each frame.

Finally a quadratic rate-distortion model is used to calculate the corresponding quantization parameter, which is then used for the RDO for each MB in the current basic unit as follows:

$$
T = \frac{x_1 \times MAD_n}{QP} + \frac{x_2 \times MAD_n}{QP^2} \tag{8}
$$

Where T is target bit rate, x_1 and x_2 are the model parameters. However, the MAD of current MB is only available after performing the RDO. This is a typical chicken and egg dilemma. Because of this, the rate control for JVT video coding is used to solve to estimate the MAD of current MB. Besides this, we also need to compute a target bit rate for the current MB and to determine the number of contiguous MBs that share the same quantization parameter.

3 New Macroblock-Layer Rate Control

In JVT-H014[5], the target bit is estimated solely based on the buffer fullness, regardless of the inter-frame motion. [7] introduced the MAD ratio as measure of motion complexity to improve the video quality at scene changes. However, MAD ratio is not a good ways for representing the motion contents, as it can only represent the similarity between the current frame and its reference frames. Also scene change, zooming, and fast motion may occur image quality degradation due to little correlation to the previous frames. This section deals with a new bit rate control scheme for H.264/AVC considering characteristics both of previous and current macroblock of frame and predicted buffer status, which controls for buffer overflow and underflow.

3.1 Bit Allocation per Frame for Target Bit Rate

In the section 2.1, we shows that JVT video coding is wrongly estimated the target bit rate for encoding frame. Then we proposed the effective predicted the target bit rate. That is to say, if estimated target bit rate is less than ‘0’, we recognized status of buffer occupancy is overflow. We encoded adding 2 from QP of previous frame(QP_{pre}) in order to both diminish the target bit rate for encoding frame and maintain the smoothness of visual quality among successive frames.

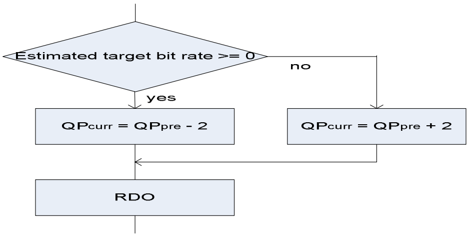


Fig. 3. Flowchart for proposed algorithm

3.2 Improved Quadratic R-D Model and Bit Allocation for Macroblock

There are two considerations to decide the quantization parameter for macroblock.

Case 1: Available bit to encode macro block in a frame is greater than zero
The quantization parameter corresponding to the target bit is then computed by using the modified quadratic mode as in *eq*. (9).

$$T_{MB(i,j)} = \frac{a_1 \times MAD'_{CB(i,j)}}{QP} + \frac{a_2 \times MAD'_{CB(i,j)}}{QP^2} \qquad (9)$$

Where, $T_{MB(i,j)}$ is the target bit rate for current macroblock to be encoded. $MAD'_{CB(i,j)}$ is mean absolute difference between previous and current frame considering not MB's pixel value of motion compensation but co-located pixel value of MB for scene change as like figure 4. Also formula of $MAD'_{CB(i,j)}$ is given in *eq*. (10). Also a_1 and a_2 is updated after encoding.

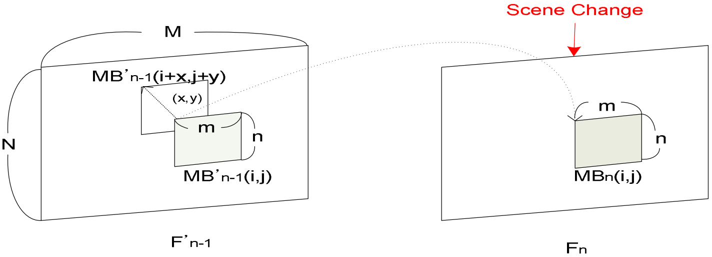


Fig. 4. Prediction of $MAD'_{CB(i,j)}$ for proposed rate control

$$\begin{aligned} MAD_n(i,j) &= \frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|MB_n(i,j) - MB'_{n-1}(i,j)\right| \\ MAD'_{CB(i,j)} &= \frac{1}{X \times Y}\sum_{i=0}^{X-1}\sum_{j=0}^{Y-1} MAD_n(i,j), (X = \frac{M}{m}, Y = \frac{N}{n}) \end{aligned} \tag{10}$$

Case 2: Available bit to encode macro block in a frame is smaller than zero

In this case the bit rate allocated to current macro block would be insufficient. Therefore picture quality of the macroblock and circumferential macroblock of the frame should be smoothness. To this end, as illustrated in figure 5, proposed algorithm adjusts the quantization parameter of the current macroblock in the same location of previous frame according to the macroblock mode of current frame. If current macro block mode is SKIP mode then the bit stream for texture and moving data will be '0'. Therefore use quantization parameter for macro block in previous frame. If current macro block mode is INTER mode bit stream will be less than in INTRA mode. Therefore add 1 to quantization parameter for macro block of previous frame. If current macro block mode is INTRA mode then add 2 to quantization parameter for macroblock of previous frame.

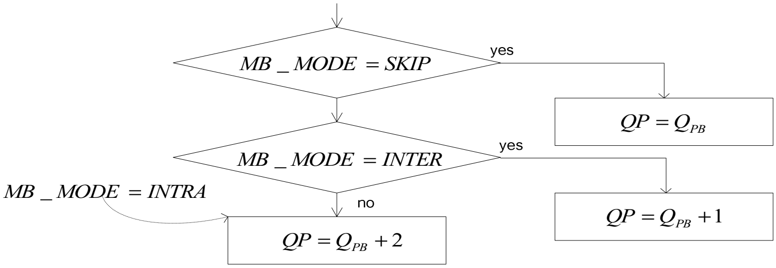


Fig. 5. Decision of quantization parameter for macroblock

3.3 Post-procession

This phase, the algorithm stores MAD which is coding result for macroblock, quantization parameter and coded bit rate(Bit_{MB}). Then it update target bit rate reflecting the coding result for next macroblock and the target bit rate for the frame as *eq*. (11).

$$T = T - Bit_{MB} \tag{11}$$

4 Experimental Results

To demonstrate the performance, we implement the proposed rate control algorithm in H.264 encoder reference software version 9.5[8]. This section presents the experimental results on typical test sequences. As recommended by JVT, the simulation common conditions presented by VCEG-N8111 are used. The test sequences are encoded by JM9.5 at the bit rate 32kbps and 64kbps. The generated bit rates are used as the target bit rates for the encoder with our rate control scheme and the one with JVT-H014 and JVT-D030 rate control scheme. For comparison, the basic unit in JVT-H014 rate control is selected to be a macroblock. In all simulation tests, the encoding parameters have been set next followings. The first picture is INTRA(I-TYPE) coded and the remaining pictures are P-TYPE. For all tests, the CAVLC mode is enabled and ME(Motion Estimation) search windows is set to 16. All other parameters such as de-blocking filter, context initialization and file mode have been carefully selected equivalent. In the test set, we used seven sequences with CCIR-601 parameters (176x144 pixels and a frame rate of 10fps): FOREMAN, CARPHONE, M&D(Mother&Daughter), SILENT, MOBILE, STEFAN, and TABLE TENNIS. In all experiments, the number of encoded frames was 100. We chose a GOP size of 10. The search was selected as 16x16 for P-type frames, and the alternate scan option was turned on for all cases.

Figure 5 depicts the estimated target bit rate and allocated bit rate for TABLE TENNIS sequence in the proposed algorithm. Proposed algorithm is efficiently distributes in the whole frames of both GOP and target bit rate in the scene change situation.

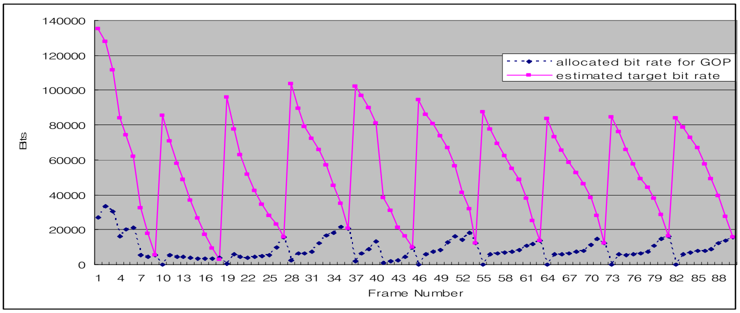


Fig. 5. Estimated target bits and allocated bit rate for TABLE TENNIS in the proposed algorithm

Table 1. The average PSNR values and average encoded bit rate at 32kbps for (I) JVT-D030, (II) JVT-H014, and (III) proposed algorithm

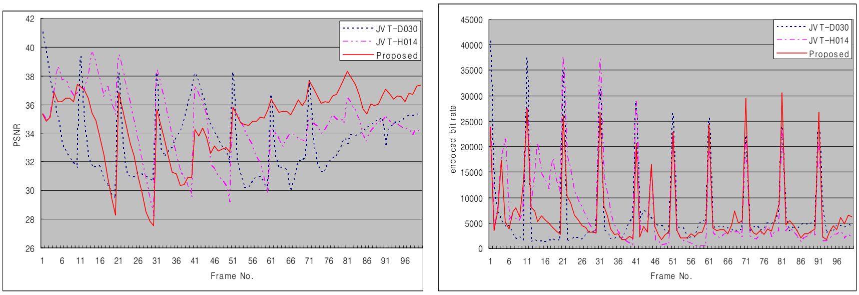
Sequence	JVT-D030		JVT-H014		Proposed	
	PSNR	encoded bit rate	PSNR	encoded bit rate	PSNR	encoded bit rate
FOREMAN	29.59	31.93	31.25	39.09	31.86	32.03
CONTAINER	34.22	31.96	35.91	41.00	35.89	31.84
M&D	36.71	31.98	37.31	39.02	37.72	31.51
SILENT	31.68	31.96	33.09	41.10	32.54	31.46
STEFAN	22.36	32.04	26.30	47.76	25.57	32.05
TABLE TENNIS	29.74	31.98	30.41	42.33	31.08	32.07
MOBILE	21.96	32.01	24.29	52.99	24.50	32.05

Table 1 and Table 2 show the PSNR performance and the encoding the bit rate of performance comparison for 1) JVT-D030, 2) JVT-H014, and 3) Proposed algorithm respectively at the 32kbps and 64kbps. The results indicate that significant PSNR gains can be obtained with the proposed algorithm. It is shown that by the proposed method the generated bit rates are very close to the target bit rates while keeping good video quality. These results justify that the proposed algorithm is superior to JVT-H014 rate control because of correctly predicting the target bit rate of each frame for scene change or high motion.

Table 2. The average PSNR values and average encoded bit rate at 64kbps for (I) JVT-D030, (II) JVT-H014, and (III) proposed algorithm

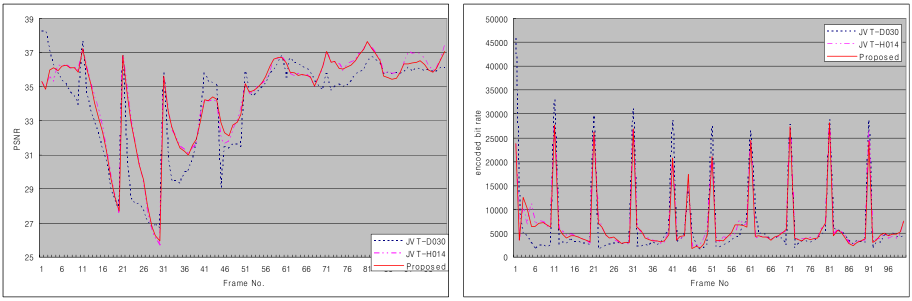
Sequence	JVT-D030		JVT-H014		Proposed	
	PSNR	encoded bit rate	PSNR	encoded bit rate	PSNR	encoded bit rate
FOREMAN	33.42	63.96	34.43	67.74	34.48	63.44
CONTAINER	37.71	63.96	38.89	78.08	38.40	63.17
M&D	40.20	63.94	40.37	67.10	40.80	63.10
SILENT	35.50	63.82	36.62	77.26	35.90	63.99
STEFAN	25.57	64.12	26.30	81.86	27.02	64.11
TABLE TENNIS	33.90	63.98	34.53	73.11	34.90	63.97
MOBILE	25.65	63.99	27.19	77.51	27.23	63.94

Figure 6 and 7 represents value of PSNR and encoded bit stream for FOREMAN and TABLE TENNIS sequences respectively at the 64kbps. In the case of encoded bit rate and PSNR, our proposed algorithm and JVT-D030 does not change abruptly in the high motion, but JVT-H014 is changed according to variance of motion.



(a) (b)

Fig. 6. (a) PSNR value at each frame and (b) encoded bit rate at each frame for FOREMAN sequence for (I) JVT-DO30, JVT-H014, and (III) proposed algorithm



(a) (b)

Fig. 7. (a) PSNR value at each frame and (b) encoded bit rate at each frame for TABLE TENNIS sequence for (I) JVT-DO30, JVT-H014, and (III) proposed algorithm

5 Conclusions

In this paper, we proposed an effective rate control algorithm for abruptly scene change and high motion by more accurately predicting frame complexity using quadratic rate-distortion the statistics of previous and current macroblock of frame. Based on our extensive tests and computational analysis, we show that the coding efficiency achieved by the proposed rate control algorithm is similar to or even better than that of the JVT-D030 and the JVT-H014 rate control algorithm while keeping high motion and scene change.

Acknowledgement

This work was supported by Korea Research Foundation Grant (KRF-2004-041-D00527).

References

1. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG Document JVT-G050r1 (2003)
2. ISO/IEC JTC1/SC29/WG11 MPEG/N0400: MPEG-2 Test Model5 (1993)
3. T. Wiegand, H. Schwarz, A. Joch, F. Lossentini, and G. J. Sullivan: Rate-constrained Coder Control and Comparison of Video Coding Standards. IEEE Transaction on Circuit and System for Video Technology, Vol. 13, No 7(2003) 688–703
4. Siwei Ma, Wen Gao, Peng Gao, Yan Lu: Rate Control for JVT Standard. JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-D030.doc(2002)
5. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG Document JVT-H014(2003)
6. Siwei Ma, Wen Gao, Peng Gao, Yan Lu: Rate Control for Advanced Video Coding standard. Proceedings of the 2003 International Symposium on ISCAS '03., Vol. 2(2003) 892-895.
7. M. Q. Jiang, X. Q. Yi, N. Ling: Improved Frame-layer Rate Control for H.264 using MAD Ratio. in Proc. 2004 IEEE Int. Symp. Circuits Syst., Vol 3(2004), 813-816
8. JVT Reference Software Version JM9.5: http://bs.hhi.de/~suehring/tml/
9. F. Pan, Z. Li, K. Lim, G. Feng: A Study of MPEG-4 Rate Control Scheme and Its Improvements. IEEE Trans. Circuit Syst. Video Technol., Vol. 10(2000) 878-894

Selective Quality Control of Multiple Video Programs for Digital Broadcasting Services

Hong-Yoen Yu, Aaron Park, Keun Won Jang, Jin Park, Seong-Joon Baek,
Dong Kook Kim, Young-Chul Kim, and Sung-Hoon Hong*

The School of Electronics and Computer Engineering,
Chonnam National University, Gwangju, South Korea, 500-757
hsh@chonnam.ac.kr

Abstract. This paper presents a selective quality control system to be able to accurately control the relative picture quality among the video programs in terms of PSNR. The proposed system allows variable bit rate (VBR) for each video program to maintain the pre-determined relative picture quality among the aggregated video programs while keeping a constant sum of the bit rates for all programs to be transmitted over a single constant bit rate (CBR) channel. This is achieved by simultaneous controlling the video encoders to generate VBR video streams at the central controller. Furthermore, we also suggest buffer regulation method based on the analysis of the constraints imposed by sender/receiver buffer sizes and total transmission rate. Through various simulation results, it is found that the proposed quality control system guarantees that the video buffers neither overflow nor underflow and the quality control errors do not exceed 0.1 dB.

1 Introduction

A conventional constant bit rate (CBR) channel is now capable of delivering several digitally compressed video programs due to advances in video compression, such as MPEG-2 [1], and digital transmission technology. Some video applications in this multi-program transmission environment are digital satellite TV, digital CATV, digital terrestrial TV, and so on. Viewers watching these video services demand higher quality services. To satisfy this demand, wider channel bandwidth should be allocated to each program. However, this means that the number of programs capable of being transmitted through the limited channel bandwidth is reduced. On the other hand, viewers may request that some programs be coded in higher picture quality than others depending on the contents of programs. Therefore, if the relative picture quality among the programs can be discriminated according to the importance of each program, more improved video services can be provided to the viewers.

The most typical method used in the multi-program transmission environment is an independent coding in which each program is CBR compressed and then

* Corresponding author.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 832–841, 2006.

transmitted at a fixed bit rate as shown in Fig. 1. However, picture complexity of a program may vary along the time, thus the independent coding scheme can not provide the consistent picture quality and the discriminated picture quality to the viewers according to the importance of the program.

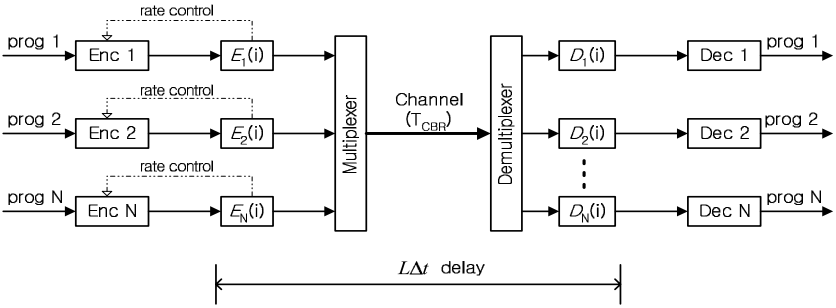


Fig. 1. Independent coding system

To get the higher quality than the independent coding, several methods [2, 3, 4] were studied. These methods tried to obtain the advantages of VBR compression, such as more consistent and higher picture quality. They allow the variable bit rate for an individual video program while maintaining a constant sum of the bit rates for all programs to be transmitted over a single CBR channel. This is achieved by simultaneously controlling the video encoders to generate the VBR video streams at the central controller. However, these methods can not control the relative picture quality among the programs according to the importance of each program, and so it is hard to satisfy the viewers demands for higher quality services. Furthermore, these methods may not guarantee to prevent buffer underflow and overflow.

In this paper, we propose a selective quality control scheme that can accurately control the relative picture quality among MPEG-2 video programs in terms of PSNR. Furthermore, we also suggest buffer regulation method based on the analysis of the constraints imposed by sender/receiver buffer sizes and total transmission rate.

2 Proposed Selective Quality Control System

Figure 2 shows the proposed selective quality control system. In this system, a single common encoder buffer with finite size is used to share in multiple video streams but each decoder has its own individual decoding buffer. The output bit-streams from the encoders are transmitted over a CBR channel after multiplexed and stored in the common buffer. At the de-multiplexer, incoming streams from the channel are identified by the PID(packet identifier) as to which stream belongs to, and they are passed to the their own decoder buffers where they await decoding by the decoder [5]. Our system allocate the variable bit rate for an individual video program to keep the pre-defined relative picture quality between the programs while maintaining a constant sum of the bit rates for all programs to be transmitted over a single CBR by simultaneously controlling all the video encoders at the central controller. Moreover, our system controls the total bits generated from all the encoders and transmission bit rate of each program so as to prevent from overflowing and undeflowing for both of encoder and decoder buffers. For this, central controller in the encoder side monitors the bit amount generated from each encoder, and determines the quantization parameter (QP) and the transmission bit amount for each program during every frame period.

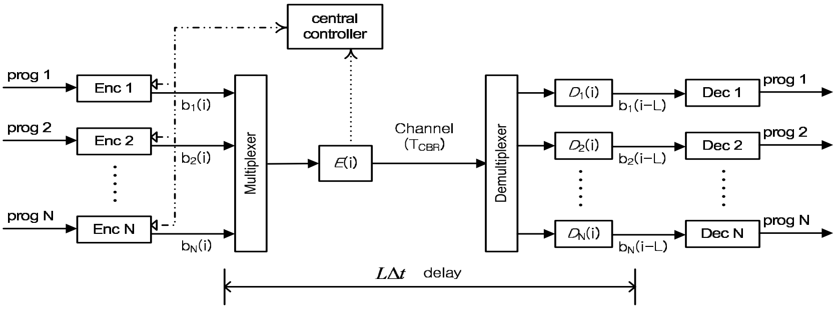


Fig. 2. Selective quality control system

We should consider the following conditions when design the selective quality control system: 1) Sum of the transmission bit rates for all the programs should be almost equal to (but not larger than) the CBR channel bandwidth. 2) Encoder and decoder buffers neither overflow nor underflow. 3) Each video program keeps the relative picture quality which is pre-determined according to the importance of the program contents.

To meet the above considerations, we adopt the rate-distortion model proposed by Hong [6] et al., which has some outstanding advantages: 1) computational complexity is small because its major operation is just to obtain a histogram or weighted histogram of DCT coefficients from an input picture, and the final formula of the estimation model is simple. 2) Estimation results are very accurate enough to be applied to practical video coding applications.

2.1 Constraints on Encoder and Decoder Buffers

In this section, we present the constraints necessary to guarantee that the buffers at the multiplexer and receivers do not overflow and underflow. Let $\triangle t$ be frame period and the interval be $[(i-1)\triangle t, i\triangle t]$. The symbols related to the buffer constraints are as follows.

N : the number of multiplexed video programs
$E(i)$: occupancy of the encoder buffer with size E^{max} at time $i\triangle t$
$b_k(i)$: bit amount generated from the k^{th} program during the interval
$B_T(i)$: total bit amount generated from all the programs during the interval
$t_k(i)$: transmission bit amount for the k^{th} program during the interval
$T(i)$: total transmission bit amount for all the programs during the interval
$D_k(i)$: occupancy of the k^{th} decoder buffer with size D_k^{max} at time $i\triangle t$

In the selective quality control system as shown in Fig. 2, the conditions to prevent encoder and the decoder buffers overflowing and underflowing are

$$0 \ \le E(i) \ \le E^{max} \tag{1}$$

$$0 \le D_k(i) \le D_k^{max} \tag{2}$$

and total transmission rate should satisfy (3) when the N VBR bit streams transmitted through a CBR channel with the bandwidth of T_{CBR} (bits/frame).

$$T(i) = \sum_{k=1}^{N} t_k(i) \le T_{CBR} \tag{3}$$

Since encoder buffer fullness can be represented by

$$E(i) = E(i-1) + B_T(i) - T(i), \tag{4}$$

the range of $B_T(i)$ to prevent encoder buffer underflow and overflow is

$$\begin{aligned} B_T(i) &\in \{\min B^e(i), \max B^e(i)\} \\ &= \{\max[T(i) - E(i-1), 0], E^{max} + T(i) - E(i-1)\}. \end{aligned} \tag{5}$$

On the other hand, buffer fullness of the k^{th} program $D_k(i)$ can be represented by

$$D_k(i) = D_k(i-1) + t_k(i) - b_k(i-L). \tag{6}$$

From (2) and (6), the range of $t_k(i)$ to prevent decoder buffer of k^{th} program underflow and overflow is

$$\begin{aligned} t_k(i) &\in \{\min t_k^d(i), \max t_k^d(i)\} \\ &= \{\max[b_k(i-L) - D_k(i-1), 0], D_k^{max} + b_k(i-L) - D_k(i-1)\}. \end{aligned} \tag{7}$$

From the above equation, we can find out the range of $t_k(i)$ is depending on only previous frame information. From (3) and (7), we can find out that $T(i)$ to prevent decoder buffers as well as to maximize the channel utilization is

$$T(i) = \min[\sum_{k=1}^{N} \max t_k^d(i), T_{CBR}]. \tag{8}$$

Using (7) and (8), we can satisfy the decoder buffer constraint by distributing the total transmission rate $T(i)$ to individual transmission rate $t_k(i)$ properly. The $t_k(i)$ is determined by following procedure during every frame period.

Step 1. calculates the average of the bound of $t_k(i)$ in (7) such that

$$avgt_k(i) = (\min t_k^d(i) + \max t_k^d(i))/2. \tag{9}$$

Step 2. calculates the estimated value of $t_k(i)$ such that

$$\hat{t}_k(i) = T(i) \times \frac{avgt_k(i)}{\sum_{j \in S} avgt_j(i)},\ for\ k \in S, \tag{10}$$

where S is a set of program numbers that their transmission rate have not been determined yet.

Step 3. If $\hat{t}_k(i) \notin \{\min t_k^d(i), \max t_k^d(i)\}$ for $k \in S$, assign the bounded value to $t_k(i)$ and then update $T(i) = T(i) - t_k(i)$.

Step 4. If $\sum_{j \in S} \hat{t}_j(i) \leq T(i)$ for $k \in S$, assign the estimated value such that $t_k(i) = \hat{t}_k(i)$ and stop the procedure. Else go to step 2.

2.2 Basic Concept of the Selective Quality Control

Figure 3 shows an example of the selective quality control when three programs are multiplexed and the quality ratio is maintained $1/\beta_1 : 1/\beta_2 : 1/\beta_3$ in terms of MSE aspect. The procedure is as follows. First, Rate-distortion (R-D) curve of each program $R_k(D)$ is modified to $R_k(D/\beta_k)$ by extending $R_k(D)$ along the distortion (horizontal) axis as much as its quality ratio β_k. Second, the global R-D curve $R_T(D)$ is constructed by adding the modified R-D curves, $R_k(D/\beta_k)$'s, of all the programs in the direction of the rate axis (vertical direction). Third, we find the target distortion D_T at which the envelope of $R_T(D)$ intersects the horizontal $R_T(D) = B_T$, where B_T is the total amount of bits to be allocated to all programs in a frame period. Finally, the bits B_k is re-allocated to the

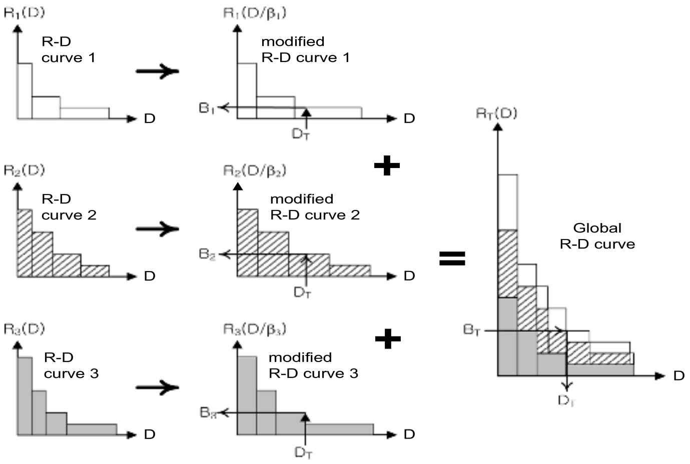


Fig. 3. Bit allocation method for selective quality control

current picture of the k^{th} program by obtaining the point at which the envelope $R_k(D/\beta_k)$ intersect the line $D = D_T$. As a result, the amount of the bits allocated to the picture of the k^{th} program is B_k and the corresponding distortion is D_T/β_k and relative distortion ratio is $1/\beta_1 : 1/\beta_2 : 1/\beta_3$.

2.3 Selective Quality Control Algorithm

Selective quality control system performs the following steps during each frame period:

Step 1. determines the relative quality ratio of all the programs to be multiplexed. When the k^{th} program keeps the picture quality with $\triangle PSNR_k$ higher than the other program with picture quality $PSNR_Q$, weighting factor β_k that is to be used in the R-D curve modification of the k^{th} program is determined by

$$\beta_k = 10^{0.1 \times \triangle PSNR_k}, \tag{11}$$

because the relationship between PSNR and MSE is

$$PSNR_Q + \triangle PSNR_k = 10 \log_{10} \frac{255^2}{MSE_Q/\beta}. \tag{12}$$

Step 2. estimates the R-D curves of each picture to be encoded by using the Hong's rate and distortion estimation formula [6] and calculates total target bits B_T to be allocated to the current pictures of all the programs.

The total target bit is determined by the following steps. First, target bits for the current pictures of every program are individually calculated, and then the total target bit is obtained by summing them. Next, this total target bit is adjusted to prevent encoder buffer overflow and underflow. The individual target bit for each programs picture is obtained by extending the target bit allocation method suggested in MPEG-2 TM5 [7]. When the picture coding type of the current picture of the k^{th} program is *pct* $\in \{I, P, B\}$, the target bit for this picture is

$$b_k^{pct} = \frac{X_k^{pct}/K^{pct}}{\sum_{pct \in \{I,P,B\}} (X_k^{pct}/K^{pct}) N_k^{pct}} R_k, \tag{13}$$

where X_k^{pct} is a complexity measure of the current picture of the k^{th} program and $K^{pct}$is a universal constant dependent on the quantization matrix. And N_k^{pct} is the number of pictures with *pct* coding type remaining in the current GOP of the k^{th} program. R_k is the remaining bits assigned to the GOP of the k^{th} program. So, total bit amount $B_T(i)$ generated from all the programs during the interval $[(i-1)\triangle t, i\triangle t]$ is calculated by summing b_k^{pct} such that

$$B_T(i) = \sum_{k=1}^{N} b_k^{pct}. \tag{14}$$

Then, the total target bits B_T is adjusted to satisfy the encoder buffer constraint shown in (5) such that

$$B_T(i) = \max\left[(1+\lambda) \times \min B^e(i), \min[B_T(i), (1-\lambda) \times \max B^e(i)]\right], \quad (15)$$

where λ is a constant numbers to prevent the buffer status from being too close to bounds in (5).

Step 3. obtains the modified R-D curves, $R_k(D/\beta_k)$'s, by multiplying the predetermined relative quality ratio β_k and the R-D curve $R_k(D)$ estimated in the step 2, and constructs the global R-D curve $R_T(D)$ by adding the modified R-D curves.

Step 4. calculates the total target distortion D_T that is corresponding to the total target bits B_T such that

$$D^T = \underset{D}{argmin}\, |R_T(D) - B_T(i)| \quad (16)$$

and selects a quantization parameter (QP) for each picture corresponding to the modified distortion estimation, and then encodes each picture with its own selected QP level. The QP level for the k^{th} program is obtained by

$$QP_k = \underset{1 \leq QP \leq 31}{argmin}\, |\hat{D}_k(QP) \times \beta_k - D_T| \quad (17)$$

where $\hat{D}_k(QP)$ is the distortion estimation value of the k^{th} program's picture. Therefore, the selected QP is the minimum value among the QP levels corresponding to the modified distortion estimation values greater than D_T.

When encoding each programs picture with its own QP selected by (17), the sum of the bits generated from the actually coded pictures of all programs may not be exactly equal to the total target bit amount $B_T(i)$. However, we can prevent encoder buffer underflow and overflow by using the constant λ used in the total target bit estimation step. Furthermore, we can prevent decoder buffer underflow and overflow by using the $t_k(i)$ determination procedure.

3 Simulation Results

To evaluate the performance of the proposed selective quality control, we carried out simulations with four standard MPEG video sequences (720×480 spatial resolution, 30 frames/sec, 4:2:0 format). The order of the multiplexed programs is *mobile, flower garden, football,* and *popple* sequences. And we used the fixed GOP structure of IBBPBBPBBPBB. The total channel rate, R_{CBR}, was set to 20 Mbps (average channel rate allocated to each program is 5Mbps). The common encoder buffer size was set to $R_{CBR} \times 300$ms (=6 Mbits) because end-to-end delay is usually set to 300ms in the video broadcasting services and the individual decoder buffer size was set to $2(R_{CBR}/4) \times 300$ms (=3 Mbits).

Figure 4 shows the independent coding result that each program is coded by TM5. In this case, since the simple programs are encoded with smaller QP

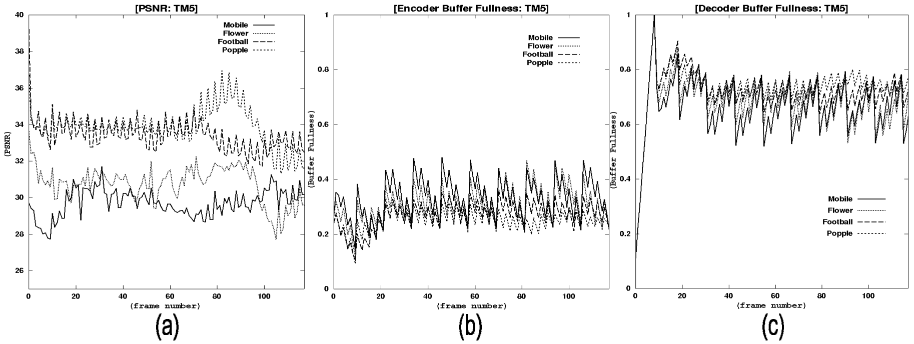


Fig. 4. Independent coding results: (a) PSNR, (b) encoder buffer status, (c) decoder buffer status

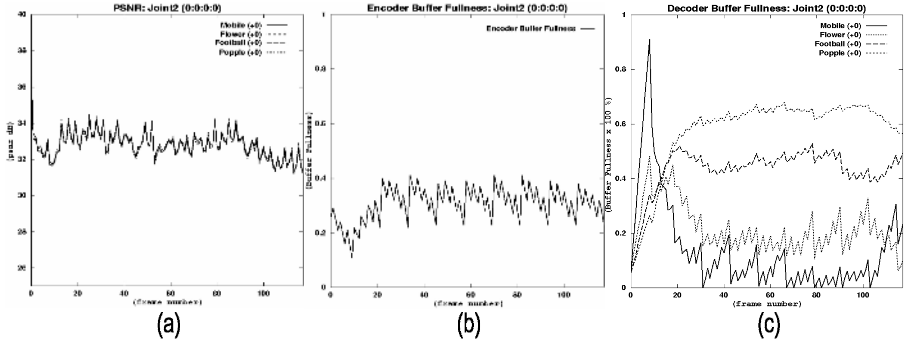


Fig. 5. Selective quality control results (△PSNR=0:0:0:0): (a) PSNR, (b) common encoder buffer status, (f) individual decoder buffer status

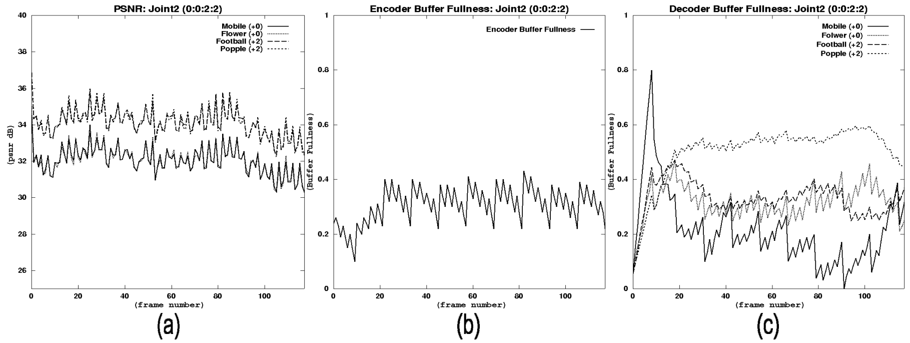


Fig. 6. Selective quality control results (△PSNR=0:0:2:2): (a) PSNR, (b) common encoder buffer status, (c) individual decoder buffer status

Table 1. Average coding results of the selective quality control

Input	PSNR=0:0:0:0			PSNR=0:0:2:2		
Seq	PSNR(dB)	b_k (Kbits)	t_k (Kbits)	PSNR(dB)	b_k (Kbits)	t_k (Kbits)
Mobile	32.84	248.08	243.09	32.11	209.90	209.89
Flower	32.83	204.69	192.25	32.10	175.54	169.66
Football	32.82	128.82	134.13	34.13	166.58	166.60
Popple	32.80	92.38	97.20	34.12	121.98	120.52
Average	32.82	168.49	166.67	33.12	168.50	166.67

levels than complex programs, large quality differences between the programs occur. And this result also shows that the quality difference between the consecutive frames is also large. That means the independent coding does not provide consistent picture quality even within a program.

Figure 5 and 6 show the selective quality control coding results when every program may have the same picture quality (i.e., quality difference ratio PSNR = 0:0:0:0) and when the four video programs are controlled so that the PSNR of the *football* and *popple* sequences may become 2dB higher than that of other sequences (i.e., quality difference ratio PSNR = 0: 0: +2: +2), respectively. From these results, we can see that the relative quality control is very accurate and fullness of the encoder and the decoder buffers always keep safe states.

4 Conclusion

In this paper, we proposed a selective quality control system to be applied to the multi-program video transmission environment, whereby several digitally compressed video programs are transmitted over a single CBR channel. Furthermore, we also suggest buffer regulation method based on the analysis of the constraints imposed by sender/receiver buffer sizes and total transmission rate. The proposed selective quality control scheme performs VBR compression for each program so as to maintain the pre-determined relative picture quality while the sum of the output bit rates of all encoders is kept constant. The selective quality control also has some desirable features. First, it can provide discriminated picture quality among the jointly coded programs to the viewers according to the importance of each program. Moreover, it improves average picture quality and keeps consistent picture quality between pictures as well as within a picture, as compared with the independent coding that performs CBR compression for each program. Therefore, the viewers will be supplied with higher quality video services. Second, satisfying the buffer constraint in the total target bit estimation step, this scheme prevents video buffer underflow and overflow.

Acknowledgments

This work was supported by the Post-doctoral Fellowship Program of Korea Science & Engineering Foundation (KOSEF).

References

[1] ISO-IEC/JTC1/SC29/WG11: Generic coding of moving pictures and associated audio information: Video. ISO-IEC 13818-2, (1994)

[2] Sakazawa, S., Takishima, Y., Wada, M., Hatori, Y.: Coding control scheme for a multi-encoder system. 6th International Workshop on Packet Video (1996) 83-88

[3] Wang, L., Vincent, A.: Bit allocation for joint coding of multiple video programs. SPIE VCIP97, (1997) 149-158

[4] Hong, S-H., Kim, S-D.: Joint Video Coding of MPEG-2 Video Programs for Digital Broadcasting Services. IEEE Trans. on Broadcasting, **44** (1998)

[5] ISO-IEC/JTC1/SC29/WG11: Generic coding of moving pictures and associated audio information: System. ISO-IEC 13818-1, (1994)

[6] Hong, S-H., Yoo, S-J., Lee, S-W., Kang, H-S.: Rate Control of MPEG Video for Consistent Picture Quality. IEEE Transactions on Broadcasting **49** (2003), 1-13

[7] Document ISO-IEC/JTC/SC29/WG11, "Test Model 5," Draft, (1993)

Joint Source-Channel Video Coding Based on the Optimization of End-to-End Distortions

Wen-Nung Lie[1], Zhi-Wei Gao[1], Tung-Lin Liu[1], and Ping-Chang Jui[2]

[1] Department of Electrical Engineering, National Chung Cheng University,
Chia-Yi, 621, Taiwan, ROC
wnlie@ee.ccu.edu.tw
[2] Materials & Electro-Optics Research Division, Chung Shan Institute of Science &
Technology, Lung-Tan, Tao-Yuan 325, Taiwan, ROC

Abstract. In this research, a JSCC (Joint Source-Channel Coding) video coding system based on the optimization of end-to-end distortions is proposed. To this end, a model describing the rate and distortion of video contents in the error-prone environment is established and estimated. Based on the constructed models, the proposed system is capable of controlling coding parameters, such as channel code rate, quantization parameters, and number of packets, adaptively in accordance with the channel condition and bit rate budge, to optimize the quality of received video at decoder side. Experimental results show that our proposed JSCC system can improve PSNR by up to 5 dB, with respect to the traditional EEP (Equal error protection) technique.

Keywords: Error resilient video coding, End-to-end distortion model, Joint source-channel video coding, Rate control, Rate compatible punctured code.

1 Introduction

Due to the development of DCT, Motion Estimation and Compensation (ME/MC), and VLC, a large amount of redundant information in video can be removed and allow the data size to be reduced for transmission over channels of limited bandwidth. However, compressed videos are vulnerable to channel errors; even a minor disturbance may make the received video bit stream undecodable. Hence, techniques such as error resilient video coding and error concealment [1], [2], were proposed to enhance the robustness of video again errors.

Another technique commonly used to enhance the quality of video transmission is channel coding. Although Shannon's separation principle provides a theoretical foundation to allow us to optimally design the source and the channel coders separately, this, however, can not be applied to real scenarios, due to the violation of assumptions, i.e., it is impossible to design an extremely long channel code to protect the transmitted information. In real applications, data transmission actually benefits from a Joint design of the Source and Channel Coding (JSCC).

Several JSCC approaches were proposed in [3], [4], [5], [6] and [7], where some coding parameters crucial to enhance the video robustness against channel errors, such as channel coding rate, number of intra-coded macroblocks (MBs), interval of inserting resynchronization markers, etc., were adaptively adjusted based on the estimated end-to-end distortions. In this way, quality of the received videos, in

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 842 – 851, 2006.

presence of channel errors, can be optimized. To this end, models considering overall distortion of video contents in error-prone transmission environment need to be established. In general, this kind of distortion can be decomposed into source distortion and channel distortion and hence can be established separately. In [3], the above concept was embodied in subband video coding. However, the process of estimating channel distortion incurs a high computing overhead, due to the simulation of various possible transmission errors. In [4] and [5], a theoretical channel distortion model was derived. By estimating the model parameters which are content adaptive, one can estimate channel distortion via model prediction. Hence, finding channel distortion is reduced to the problem of model fitting and the computing overhead can be significantly lowered. However, channel distortion models provided in [4] and [5] might not be accurate enough, due to the simplification of error propagation caused by lost MVs. In [6], [7] and [8], the effect of error propagation caused by lost MVs is considered in their overall end-to-end distortion models. However, the lack of rate-distortion behavior of the video source encoder in their models makes them useless in optimizing the quantization parameters.

In this paper, a more accurate channel distortion model than the one introduced in [4] and [5] was proposed by considering the effect of error propagation caused by lost MVs. This channel distortion model is then combined with the source R-D model, modified from that in [9] and [10], to develop a JSCC coding system. The proposed coding system is capable of adaptively adjusting coding parameters, including frame quantization parameters, number of slices in a frame, i.e., here, a slice means rows of consecutive MBs, and channel code rate for each slice, to yield an optimally received video quality in presence of channel errors.

2 The Source R-D Model

An R-D model is a mathematical formula used to describe the mutual behavior between the rate and distortion for a coding algorithm and a coded content. This model can be used to predict the corresponding source distortion at a given source bit rate, or vice versa. Several source models had been proposed [9], [10]. In this paper, the TMN8 model for H.263 in [9] is adopted and further modified to be able to accommodate R-D behavior of AVC/H.264 video encoder more precisely. The source R-D model adopted is:

$$\begin{cases} B_i = A\left(K\dfrac{\sigma_i^2}{Q_i^2}+C\right) & ,\dfrac{\sigma_i^2}{Q_i^2} > \dfrac{1}{2e} \\ B_i = AC & ,otherwise \end{cases} , \tag{1}$$

$$D_i = \frac{\gamma\sigma_i^2 + \eta}{12} e^{-\alpha \bar{b}_i} ,\ \alpha = 2\ln 2 , \tag{2}$$

where B_i is the bit rate consumed by the i^{th} MB; A is a constant representing the number of pixels in a MB; C represents the average overhead of header information for a MB; σ_i^2 represents the variance of residual of the i^{th} MB; K is a constant; Q_i is the quantization factor; D_i is the distortion of the i^{th} MB, $\bar{b}_i$ represents the average number of bits per pixel consumed by the i^{th} MB, and γ and η are constants.

The parameters, K, C, γ and η, are content-dependent and hence need to be estimated for each frame (or for a short time interval in a sequence). Our system first pre-encodes a frame by using a pre-determined quantization factor (here, a pre-determined value according to the allowable bit rate) to collect data points, such as B_i, σ_i^2, D_i and $\overline{b_i}$. Then, the optimal values of K, C, γ and η that best fit the model in Eqs.(1) and (2) in the least squared error sense can be solved for later use in JSCC optimization.

3 The Channel Distortion Model

Essentially, the end-to-end distortion for videos transmitted in error-prone environments can be decomposed into parts of source distortion D_s, incurred by n_q, and channel distortion D_c, incurred by n_c [4], [5]. Here, n_q is related to the quantization noise and n_c to the incompleteness of error concealment and motion compensation. A pictorial illustration of our model is depicted in Fig. 1, where f_n represents the original video signal, $\hat{f}_n$ represents the encoded video (i.e., decoded video without error), and $\tilde{f}_n$ represents the video reconstructed at the decoder side in presence of channel noise n_c. Notice that n_c is related to the strategy of error concealment applied at the decoder, as well as errors propagated via motion compensation (MC).

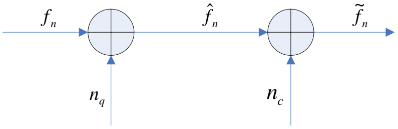


Fig. 1. The formation of end-to-end distortion

Here, the zero-motion scheme is assumed to be adopted by decoders for error concealment, that is, an erroneous macroblock (MB) is replaced by the MB at the same location of the previous frame. Since the behaviors of error propagation for intra-coded and inter-coded MBs are quite different, in the following, two channel distortion models are separately constructed accordingly.

3.1 Channel Distortion Model for Intra-coded MBs

For intra-coded MBs, channel distortion $D_{C,i}^I$ totally results from error concealment. Equation (3) below formulates this observation:

$$D_{C,i}^I = p \cdot E\left\{\left(\hat{f}_n^i - \tilde{f}_{n-1}^i\right)^2\right\}, \tag{3}$$

where i is the pixel index in an MB, n is the frame index, p is the error probability for a pixel which is closely related to the channel condition and channel code adopted by the system, $E(.)$ is the expectation operator. Equation (3) can be further arranged to yield Eq.(4):

$$\begin{aligned} D^I_{C,i} &= p \cdot E\left\{ \left(\hat{f}^i_n - \hat{f}^i_{n-1} + \hat{f}^i_{n-1} - \tilde{f}^i_{n-1}\right)^2 \right\} \\ &\le p \cdot E\left\{ \left(\left|\hat{f}^i_n - \hat{f}^i_{n-1}\right| + \left|\hat{f}^i_{n-1} - \tilde{f}^i_{n-1}\right|\right)^2 \right\} \\ &\le p \cdot E\left\{ \left(\hat{f}^i_n - \hat{f}^i_{n-1}\right)^2 \right\} + 2p \cdot E\left\{ \left|\hat{f}^i_n - \hat{f}^i_{n-1}\right| \left|\hat{f}^i_{n-1} - \tilde{f}^i_{n-1}\right| \right\} \\ &\qquad + p \cdot E\left\{ \left(\hat{f}^i_{n-1} - \tilde{f}^i_{n-1}\right)^2 \right\} \end{aligned} \tag{4}$$

where $\left|\hat{f}^i_n - \hat{f}^i_{n-1}\right|$ stands for the frame difference after encoding, which is independent of the channel condition (hence, deterministic with respect to the *E*(.) operator). Differing from [4], which assumes independence between the frame difference $\left|\hat{f}^i_n - \hat{f}^i_{n-1}\right|$ and channel distortion $\left|\hat{f}^i_{n-1} - \tilde{f}^i_{n-1}\right|$ and ignores the middle term in Eq.(4), we further arrange Eq.(4) to become:

$$\begin{aligned} D^I_{C,i} &\le p \cdot E\left\{\left(\hat{f}^i_n - \hat{f}^i_{n-1}\right)^2\right\} \\ &+ 2p \cdot \left|\hat{f}^i_n - \hat{f}^i_{n-1}\right| \cdot \left|\hat{f}^i_{n-1} - E\left\{\tilde{f}^i_{n-1}\right\}\right| \cdot a + p \cdot E\left\{\left(\hat{f}^i_{n-1} - \tilde{f}^i_{n-1}\right)^2\right\} \end{aligned} \tag{5}$$

where a=1 for intra-coded MBs (explained after Eq.(8)).

3.2 Channel Distortion Model for Inter-coded MBs

Different from intra-coded MBs where distortions are only caused by error concealment, distortions resulting from error propagation should be taken into consideration for pixels of inter-coded MBs.

First, let us consider the case in which MVs and residuals $\hat{e}^i_n$ are received correctly. In this case, the decoded pixel $\tilde{f}^i_n$ will be

$$\tilde{f}^i_n = \hat{e}^i_n + \tilde{f}^k_{n-1}, \tag{6}$$

where $\tilde{f}^k_{n-1}$ stands for the pixel value compensated from the *k*-th pixel of the frame *n*-1. If errors occur, error concealment will make $\tilde{f}^i_n$ replaced with $\tilde{f}^i_{n-1}$ in the previous frame. Combining this observation with Eq.(6), the channel distortion for pixels of inter-coded MBs can be formulated as

$$D^P_{C,i} = p \cdot E\left\{\left(\hat{f}^i_n - \tilde{f}^i_{n-1}\right)^2\right\} + (1-p) \cdot E\left\{\left[\hat{f}^i_n - \left(\hat{e}^i_n + \tilde{f}^k_{n-1}\right)\right]^2\right\}. \tag{7}$$

Similar to the derivation of Eq.(5), the first term of Eq.(7) can be arranged to yield

$$\begin{aligned} D^P_{C,i} &\le p \cdot E\left\{\left(\hat{f}^i_n - \hat{f}^i_{n-1}\right)^2\right\} + 2a \cdot p \cdot \left|\hat{f}^i_n - \hat{f}^i_{n-1}\right| \cdot \left|\hat{f}^i_{n-1} - E\left\{\tilde{f}^i_{n-1}\right\}\right| \\ &+ p \cdot E\left\{\left(\hat{f}^i_{n-1} - \tilde{f}^i_{n-1}\right)^2\right\} + (1-p) \cdot E\left\{\left(\hat{f}^k_{n-1} - \tilde{f}^k_{n-1}\right)^2\right\} \end{aligned} \tag{8}$$

where $0 \le a \le 1$. The estimation of Eqs.(5) and (8) is simply upper bounds. It needs a factor *a* for correcting the over-estimated 2nd term, especially for high-motion video whose channel distortion increases as frame difference increases. The higher the motion is, the smaller the value of *a* has.

Both Eqs.(5) and (8) constitute the channel distortion model in our system to estimate the end-to-end distortion. Once the error probability *p* and the encoding results of the previous frames are known, the expectation of channel distortion for the current frame can be estimated. The overall distortion of a frame can then be estimated by using Eq.(9).

$$D_C(n) = \sum_{i \in \text{intra-pixel}} D_{C,i}^I + \sum_{j \in \text{inter-pixel}} D_{C,j}^P \tag{9}$$

3.3 Other Issues About Channel Distortion Model

Several terms in the proposed channel distortion model need more explanations, including the error probability *p* and the first moment $E\{\tilde{f}_{n-1}^i\}$ used in Eqs.(5) and (8).

First, the determination of *p* is based on the premise that the compressed video is partitioned into slices, or packets, for transmission; one frame may contain several slices. Under the binary symmetric and additive white Gaussian noise channel and the assumption that the whole packet is discarded or lost if at least one bit error cannot be recovered, *p* can be formulated as

$$p(r, L_S) = 1 - (1 - p_e(r))^{L_s}, \tag{10}$$

where L_s is the length (in terms of bits) of a slice, $p_e(r)$ is the bit error rate (BER) after channel decoding whose rate is *r*. In this paper, the Rate Compatible Punctured Convolutional Code (RCPC) is adopted for channel coding. The value of $p_e(r)$ can be theoretically determined by knowing the channel SNR and the channel code rate *r*.

More details about the estimation of $E\{\tilde{f}_{n-1}^i\}$ can be found in [8]. Here, only the results are summarized. In case of pixels of intra-coded MBs, Eq.(11) is applied, otherwise, Eq.(12) is adopted.

$$E\{\tilde{f}_n^i\} = (1-p) \cdot \hat{f}_n^i + p \cdot E\{\tilde{f}_{n-1}^i\} \tag{11}$$

$$E\{\tilde{f}_n^i\} = (1-p) \cdot \left(\hat{e}_n^i + E\{\tilde{f}_{n-1}^k\}\right) + p \cdot E\{\tilde{f}_{n-1}^i\} \tag{12}$$

Note that $E\{\tilde{f}_{n-1}^i\}$ used in Eqs.(5)(8) should be updated according to Eqs.(11)(12) for next frame (i.e., the n^{th}). For *n*=0, we set $E\{\tilde{f}_0^i\} = \hat{f}_0^i$, i.e., no channel errors are assumed.

Though models proposed in [8] can also be applied to estimate the overall end-to-end distortion directly, the lack of rate-distortion behavior for video source encoder in their models makes them difficult to be used in the design of a JSCC system. This is because in most JSCC systems, models describing this behavior are crucial in compromising between coding efficiency and error resiliency under a given bit rate budget.

To verify the accuracy of our proposed channel distortion model, an objective measure e_{D_C} is defined below:

$$e_{D_C} = \frac{1}{N}\sum_{n=1}^{N}\left|PSNR(D_C^{(n)}) - PSNR(D'^{(n)}_C)\right|, \tag{13}$$

where N is the number of frames. Notice that $PSNR(D_C^{(n)})$ is obtained via simulation, while $PSNR(D'^{(n)}_C)$ is obtained based on the models proposed here.

Our model is compared with that in [4]. The experimental results are listed in Table 1, where the Rate Compatible Punctured Convolutional Code (RCPC) is adopted for channel encoding and the bit error rate after channel decoding is assumed to be $10^{-4.52}$. It is clearly that our model significantly improves the estimations.

Table 1. Comparison between our channel model and [4]

e_{D_C} Sequence	Model in [4]	Proposed
Foreman	3.922 dB	0.784 dB
Silent	3.630 dB	1.489 dB
Susi	3.904 dB	1.898 dB
Carphone	3.512 dB	1.126 dB
Table tennis	2.852 dB	1.911 dB
Salesman	5.292 dB	2.458 dB
Mother& Daughter	3.743 dB	1.067 dB

4 Optimized JSCC System

In our proposed JSCC system, coding parameters to be determined are listed below.

$$qp \in \boldsymbol{Q} = \left\{q_i \mid q_{i+1} = q_i + 1, 1 \le i \le \overline{N}, q_{\overline{N}/2} = q_{init}\right\} \tag{14}$$

$$r_i \in \{8/9, 8/10, 8/12, 8/14, 8/16, 8/18, 8/20, 8/22, 8/24\} \tag{15}$$

$$ns \in \{1, \cdots, N_S\} \tag{16}$$

where *qp* is the quantization factor for each frame whose range is defined via q_{init} and $\overline{N}$, r_i is the channel coding rate for each slice and *ns* represents the number of slices (maximum: N_S) used to partition a frame. The channel code adopted here is RCPC. Notice that in real applications, schemes of frame-layer rate control might be used to determine the initial quantization factor q_{init}.

With the coding parameters defined above, our problem is to, given the channel state information and bit rate budget, determine appropriate *qp* and *ns* for each frame and appropriate r_i for each slice, so as to minimize the video end-to-end distortions. The problem is formulated as follows.

$$\min_{\mathbf{Z}=\{qp,ns,r_1\sim r_{ns}\}}\left\{\sum_{i=1}^{ns}\left[D_S^i(qp) + D_C^i\left(r_i, L_S^i(ns,qp)\right)\right]\right\}$$
$$\text{Subject to} \quad \sum_{i=1}^{ns}\frac{L_S^i(ns,qp)}{r_i} \le R_{budget} \tag{17}$$

where D_S^i and D_C^i represent the source and channel distortion for the i^{th} slice, respectively, L_S^i represents the length of the i^{th} slice in bits after source coding, r_i is the channel code rate for the i^{th} slice, and R_{budget} is the bit rate budget (including both the source and channel bit rates, might be even among frames or specified by another rate control scheme) for a frame. Based on the principle of Lagrangian relaxation, the above problem becomes:

$$\min_{\mathbf{Z}} J = \min_{\mathbf{Z}=\{qp,ns,r_1\sim r_{ns}\}} \left\{ \sum_{i=1}^{ns} \left[D_S^i(qp) + D_C^i\left(r_i, L_S^i(ns,qp)\right)\right] + \lambda \cdot \sum_{i=1}^{ns} \frac{L_S^i(ns,qp)}{r_i} \right\}, \tag{18}$$

where λ is the Lagrangian multiplier. The computational complexity of the problem defined in Eq.(18) can be reduced by exploring the fact that distortions and bit rates among slices in a frame are independent. Hence, the problem with less computational complexity could be obtained as follows.

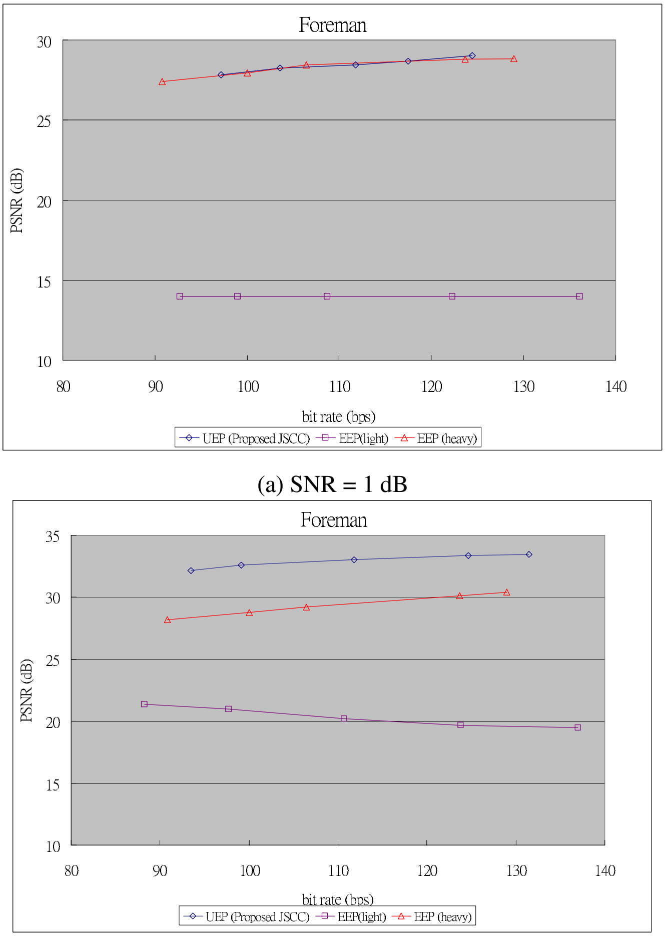
$$\min_{\mathbf{Z}} J = \min_{\{qp,ns\}} \left\{ \sum_{i=1}^{ns} \min_{r_i} \left\{ D_S^i(qp) + D_C^i\left(r_i, L_S^i(ns,qp)\right) + \lambda \cdot \frac{L_S^i(ns,qp)}{r_i} \right\} \right\} \tag{19}$$

The procedures to solve Eq.(19) are as follows.

- *STEP 1:*Values of R_{budget} and q_{init} are determined first. Empirically, the bit rate for I frames will be set to 2.5 times that for P frames. q_{init} of the 1st frame is determined according to the system bit rate, while that of other frames follows the optimized *qp* of the previous frame .
- *STEP 2:* Pre-encode the input frame by using q_{init}. After pre-encoding, information below can be collected and applied to construct accurate source and channel models:
  (1) number of bits and distortion for each MB, for estimating *K*, *C*, γ and η in Eqs.(1)(2);
  (2) MVs, residuals, and $\left(\hat{f}_n - \hat{f}_{n-1}\right)$ for each MB, for constructing the channel model in Eqs.(5)(8).
- *STEP 3:*Based on channel SNR and data collected in step 2, the source model and the channel distortion model can be established. These models are substituted into Eq.(19) to find the optimal coding parameters.
- *STEP 4:* Encode the n^{th} frame by using coding parameters obtained in step 3. Before proceeding to next step, update and save $E\left\{\tilde{f}_n^{\,i}\right\}$ and $E\left\{\left(\hat{f}_n - \tilde{f}_n\right)^2\right\}$ for each pixel position.
- *STEP 5:*Encode the next frame by repeating steps 1~4.

5 Experimental Results

Our proposed JSCC algorithm actually belongs to the UEP schemes since the channel code rate r_i for each slice may be different. Our algorithm is compared with two other EEP algorithms: light EEP and heavy EEP. The former always adopt *r*=8/9 and the later always selects *r*=8/24.



(a) SNR = 1 dB

(b) SNR = 7 dB

Fig. 2. PSNR performance of our proposed JSCC and EEP algorithms for “Foreman” sequence

The environments of experiments are: QCIF, 30 fps, 150 frames, IPPP…, 5 reference frames, RDO OFF, and Hadamard ON. In EEP, the number of slices is fixed to 2, while in our UEP algorithm, it is optimally determined and hence adaptive to variation of transmission conditions.

The results of the sequence, “Foreman”, under channel SNR = 1, 7 dB are illustrated in Fig.2. It is observed that our proposed JSCC algorithm is more adaptable than the EEP algorithms. When the channel is extremely noisy, i.e., SNR=1, the light EEP is too weak against channel errors to provide enough protection for the transmitted video. Hence, a large amount of channel distortions degrades the quality of the reconstructed video. On the other hand, when the channel is good, e.g., SNR =7, the heavy EEP is inefficient in utilizing the bandwidth available for video.

Table 2. PSNR performance under a bit rate of 300 kbps

Sequence	SNR	1 dB	3 dB	7 dB
Foreman	UEP	30.9	32.7	35.5
	Light EEP	16.4	15.9	26.7
	Heavy EEP	30.9	30.9	31.1
Salesman	UEP	32.8	34.5	37.7
	Light EEP	24.3	24.6	34.5
	Heavy EEP	32.5	32.5	32.7
Silent	UEP	31.4	33.9	36.8
	Light EEP	23.9	24.2	30.5
	Heavy EEP	31.7	32.1	32.3

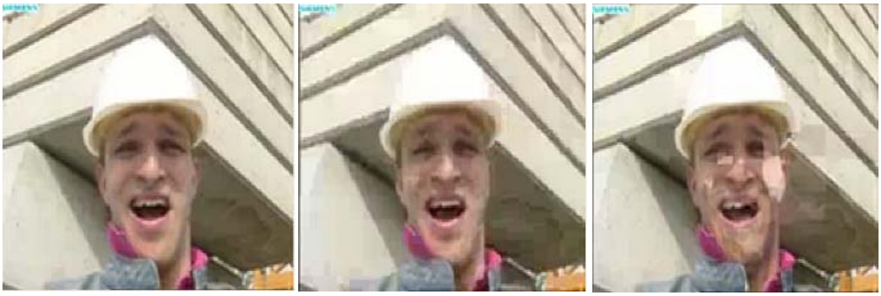

(a) proposed UEP (b) heavy EEP (c) light EEP

Fig. 3. Subjective results for different algorithms. The channel SNR is 7 dB.

Table 2 shows the results of three test sequences: “Foreman”, “Salesman” and “Silent” at a system bit rate of 300k bps. The higher the channel SNR is, the larger the improvement is (up to 5 dB, relative to the best one of light EEP and heavy EEP).

6 Conclusions

Here, a JSCC algorithm based on the end-to-end distortion model is proposed. The end-to-end distortion is decomposed into source and channel distortions. Based on the proposed models, an optimization algorithm by using Lagrangian relaxation can be applied to find a set of parameters, such as quantization parameters, number of slices in a frame, and channel code rate for each slice, that maximizes the quality of received videos. From experiments, it is observed that the proposed JSCC algorithm is adaptable to varying channel conditions and better in PSNR by up to 5 dB than the traditional EEP algorithm.

References

1. Yao Wang, Stephan Wenger, Jiangtao Wen, and Aggelos K. Katsaggelos, “Error Resilient Video Coding Techniques,” *IEEE Signal Processing Magazine***,** vol. 17, pp.61–82, 2000.
2. Yao Wang and Qin-Fan Zhu, “Error Control and Concealment for Video Communication: A Review,” *Proc. of the IEEE*, vol. 86, no. 5, pp. 974 – 997, May 1998.

3. Murari Srinivasan, and Rama Chellappa, “Adaptive source-channel subband video coding for wireless channels,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 9, pp. 1830-1839, 1998.
4. Zhihai He, Jianfei Cai, and Chang Wen Chen, “Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding,” *IEEE Trans. on Circuits and Systems for Video Tech.*, vol.12, no. 6, pp.511-523, 2002.
5. Klaus Stuhlmuller, Niko Farber, Michael Link, and Bernd Girod, “Analysis of video transmission over lossy channel,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012-1032, June 2000.
6. Yiftach Eisenberg, Fan Zhai, Thrasyvoulos N. Papppas, Randall Berry, and Aggelos K. Katsaggelos, “VAPOR: Variance-aware per-pixel optimal resource allocation,” *IEEE Trans. on Image Processing*, vol. 15, no 2, pp. 289-200, Feb. 2006.
7. Guy Cote, Shahram Shirani, and Faouzi Kossentini, “Optimal mode selection and synchronization for robust video communication over error-prone networks,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 952-965, June 2000.
8. Rui Zhang, Shankar L. Regunathan and Kenneth Rose, “Video coding with optimal inter/intra-mode switching for packet loss resilience,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no 6, pp. 966-976, June 2000.
9. Jordi Ribas-Corbera and Shawmin Lei, “Rate control in DCT video coding for low-delay communications,” *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 9, no 1, pp. 172~185, Feb. 1999.
10. Z. He, Y. Kim, and S. K. Mitra, “Object-level bit allocation and scalable rate control for MPEG-4 video coding,” *Proc. of Workshop and Exhibition on MPEG-4*, San Jose, June 2001.

A Temporal Error Concealment Method Based on Edge Adaptive Boundary Matching

Hyun-Soo Kang[1], Yong-Woo Kim[2], and Tae-Yong Kim[2]

[1] School of ECE, ChungBuk National University, Chungju, Korea
hskang@cbnu.ac.kr
[2] Graduate School of AIM, Chung-Ang University, Seoul, Korea
dawnsky77@wm.cau.ac.kr, kimty@cau.ac.kr

Abstract. In this paper, we propose a temporal error concealment method based on edge connectivity to neighboring regions through a damaged region. In the method, four regions around the damaged block, top, bottom, left, and right, are defined and the edge features of the regions are extracted by applying an edge operator for each direction. The mask for the boundary matching is determined by the edge information for each boundary region, which can be considered as a criterion to measure the activity of the boundary region. In other words, it is determined such that the mask size is proportional to the edge strength of each region in order to yield the higher reliability on boundary matching. This process is equivalent to applying weights depending on the edge features, which produces better motion vector for concealment. In experiments, it is verified that the proposed method outperforms the conventional methods in terms of image quality.

1 Introduction

While the motion prediction and the variable length coding techniques give great amount of compression in image coding, data loss at a point brings up error propagation to the following data and in the end causes serious image quality degradation.

Many techniques have been reported to relieve this problem. They may be classified to three groups: 'forward' techniques, 'post-processing' or 'concealment' techniques, and 'interactive' techniques [1]. The forward techniques are to add some redundancies for error robustness to bitstream in the encoder. These techniques have a good feature that they don't need any further error-handling in the decoder, though providing lower compression efficiency. Forward-error correction (FEC), layered coding, multiple-description coding, and transport-level control are included in these techniques. The concealment techniques are that the decoder attempts to conceal errors without supplementary information from the encoder. They utilize the spatial and temporal correlation of images for concealment. In the meantime, the interactive techniques try error recovery based on the interaction between the encoder and the decoder. Therefore, they are applicable under the condition that a feedback channel is available.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 852–860, 2006.

The concealment techniques may be divided into two classes according to the domain applying concealment techniques: spatial domain techniques and temporal domain techniques. The spatial techniques conceal the damaged region with spatially neighboring pixels regarding the high spatial correlation of image signals. The temporal techniques conceal the damaged region with temporally neighboring pixels regarding the high temporal correlation and movement of objects within images.

In this paper we propose a temporal error concealment technique based on boundary matching algorithm (BMA) using masks adaptive to luminance activity and edge information of neighboring regions. The proposed method is based on the fact that the details within objects would be continuous over damaged region and its boundary region. That is, there exists the connectivity between the damaged region and correctly decoded region in the vicinity of the damaged region in terms of the texture of an object in these regions. For example, if a single object covers the regions, there will be the connectivity over these regions. Based on this idea, we introduce the adaptive mask for boundary matching and show the performance improvement of our scheme.

2 Background

In the temporal concealment techniques, a damaged block is replaced with the block in the previous frame which is likely to be most similar to the damaged block by an error criterion such as MSE. Actually, the process to search a similar block corresponds to recovering the motion vector of the damaged block. As a result, the concealment performance absolutely depends on accuracy of the recovered motion vector.

The temporal replacement (TR) method is the simplest approach of temporal concealment in which the damaged block is replaced with the block at the same position in the previous frame, i.e. the motion vector of the damaged block is assumed to be zero [2]. Accordingly, it works well for stationary and quasi-stationary areas like background, but fails for fast moving areas.

Another technique is to replace the damaged motion vector with the average or the median of the vectors of the neighboring blocks, which is called AV method. It works well for areas with smooth motion, but fail for areas with unsmooth, e.g. object moving in different directions [3][4].

Boundary matching (BM) techniques first choose several candidate vectors from available neighboring motion vectors and find the best one to minimize the luminance difference between the boundary pixels of the damaged block and the boundary pixels of the blocks indicated by the candidate vectors [2]. These methods work well for regions with high spatial correlation. And, Iterative weighted BMA (IWBMA) was developed as an enhanced version of BMA. It iteratively applies the BMA technique updating the weights to boundary pixels in the matching process [6].

The motion field interpolation (MFI) technique finds motion vectors at control points, instead of blocks, and concealment process is performed in the unit of the points. This technique provides a smoothly varying motion field and reduces blocking artifacts [1]. The motion vector extrapolation (MVE) technique uses the motion vector obtained by extrapolation from motion vectors of blocks in two previous frames. Since one of the previous frames is temporally far from the current frame, unsmooth motion in temporal direction degrades its performance [7].

The recursive half macroblock (RHMB) matching technique consists of four steps. In the first two steps, a damaged MB is divided into upper half MB and lower half MB for which error concealment is performed separately. In the last two steps, error concealment processes for two half MB are repeated based on the cost that is obtained by applying different weights according to matched pixel positions [8].

3 Proposed Method

The proposed method attempts to use the edge information in images for error concealment. Actually, we use the strength of edges in the boundary regions adjacent to a lost block. Based on the edge information, we determine the matching mask in four directions: above, below, left and right, if available. First of all, we introduce the BM technique in more detail since the proposed method is based on the technique.

3.1 Boundary Matching Algorithm

Lam et al proposed the boundary matching algorithm (BMA) where a damaged block is searched in the previous frame with boundary matching process [2]. This process is to measure similarity between boundary pixels of the damaged block and pixels in the previous frame. Since boundary pixels of the damaged block are not available, they are predicted from the pixels just neighboring to the damaged block, which is quite acceptable due to high spatial correlation.

Consider a damaged block of $N \times N$ whose most upper-left position is (p, q). D_L, D_R, D_A, and D_B are defined as four differences between the motion compensated pixels and the boundary pixels of the damaged block in four directions: above, below, left and right. That is,

$$D_L = \sum_{i=0}^{N-1} (\hat{f}(p+i, q) - f(p+i, q-1))^2 \tag{1}$$

$$D_R = \sum_{i=0}^{N-1} (\hat{f}(p+i, q+N-1) - f(p+i, q+N))^2 \tag{2}$$

$$D_A = \sum_{i=0}^{N-1} (\hat{f}(p, q+i) - f(p-1, q+i))^2 \tag{3}$$

$$D_B = \sum_{i=0}^{N-1} (\hat{f}(p+N-1, q+i) - f(p+N, q+i))^2 \tag{4}$$

where $\hat{f}(x, y)$ denotes the motion compensated pixel value from the previous frame. The BMA chooses the best motion vector among candidate vectors which produces the smallest total difference $D = D_L + D_R + D_A + D_B$. The the candidate vectors consist of the motion vectors of (1) the block located at the same position as the current block (2) the available neighboring blocks (3) the median of the available neighboring blocks (4) the average of the available neighboring blocks (5) the zero motion vector.

The BMA works well when there is no large change in pixel values across the block boundary. However, it does not work well when neighboring pixel values change abruptly at edges or corners. To solve this problem, J. Feng proposed the modified BMA (M-BMA) to consider edge directions [5]. In the M-BMA, the boundary pixels of the damaged block are predicted from one of three directions: diagonal, anti-diagonal, and horizontal or vertical. In other words, the boundary pixels of the damage block can be predicted from two diagonal directions as well. It is clear that the M-BMA is superior to the BMA since the M-BMA consider more directions for the prediction of the boundary pixels. Even though the M-BMA considered more directions, both the BMA and the M-BMA have a weakness to replace the boundary pixels of the damaged block with neighboring pixels, which inevitably includes prediction error and may be the best in case of an objects being shrunk. To solve this problem we propose a new method where the pixels just adjacent to the damaged block are stayed at their own positions during matching process. That is, matching process is performed for their own positions but for the boundary of the damaged block. In addition, in order to keep edge connectivity between inside and outside of the damaged block, the masks adaptive to edge information are employed. The details of the proposed method will be described in the next subsection.

3.2 Edge Adaptive Boundary Matching Algorithm

Edge adaptive boundary matching algorithm (EA-BMA) is based on the fact that there may be edge connectivity between a damaged region and neighboring region as long as these regions covers an object. The pixels around a damaged block may have information about the texture of the object in the damaged region. Thus we can expect a good concealment if we use the edge connectivity between the damaged region and the neighboring region. To be sure of the connectivity, we give higher weight to the regions with the high activities or strong edges. In the end, the higher weight is realized to wider mask in matching of the direction and the lower weight is done to narrower mask.

The examples of the connectivity are shown in Fig. 1 where the line connectivity can be used as the important feature in searching for the lost block. To ensure the connectivity we prefer to include the regions with strong edges in matching mask rather than with weak edges. After all, we evaluate the edge strength of neighboring regions. The strength is employed to decide the size of the mask in each direction. This is the point different from the conventional MBA.

To measure the edge strengths of four neighboring regions, we use the Sobel operator which is applied to the region three pixels wide. The horizontal operator

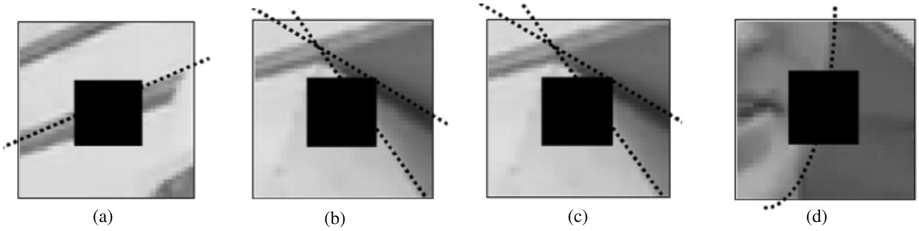


Fig. 1. Edge connectivity

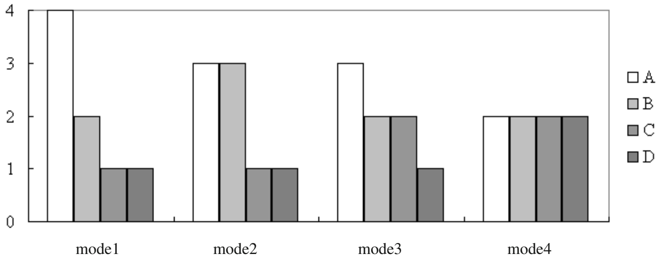


Fig. 2. Four modes for decision of the mask width

is applied to left and right adjacent regions to extract horizontal edges while the vertical operator is applied to above and below adjacent regions. We should note that the operators with a single direction were applied since vertical edges in the left and right adjacent regions do probably not pass through the lost block. The edge components resulted from the edge operation sums up to obtain the edge strengths for four neighboring regions. After that, we sort the strengths in the decreasing order. We will call four regions A, B, C, D where the region A has the strongest edge and the region D has the weakest edge, i.e. the edge strengths have the relationships of $E(A) \geq E(B) \geq E(C) \geq E(D)$, where $E(\cdot)$ denote the edge strength and $A, B, C, D \in \{\text{above, below, left, right}\}$.

Now we decide the size of the mask to be used in matching process according the edge strengths. Considering computational burden, we constrain the total widths of four masks to be 8, two pixels wide for each mask on average. To do this, four modes are introduced as shown Fig. 2. The mode is determined so that the width of the mask may be proportional to the edge strength. For instance, we select the first mode in Fig. 2 in case that $E(A) > 2E(B)$ and $E(B) > 2E(C)$. If the region is in a previously lost another block, we exclude the region in matching process to improve reliability.

Fig. 3 shows the example of the proposed mask decision. In this example, mode 1 was decided as the best mode and finally four pixel wide mask for left boundary region, two pixel wide mask for right boundary region, and one pixel wide mask for above and below regions boundary. Using the masks determined by the above procedure, we perform the matching process to obtain the following differences,

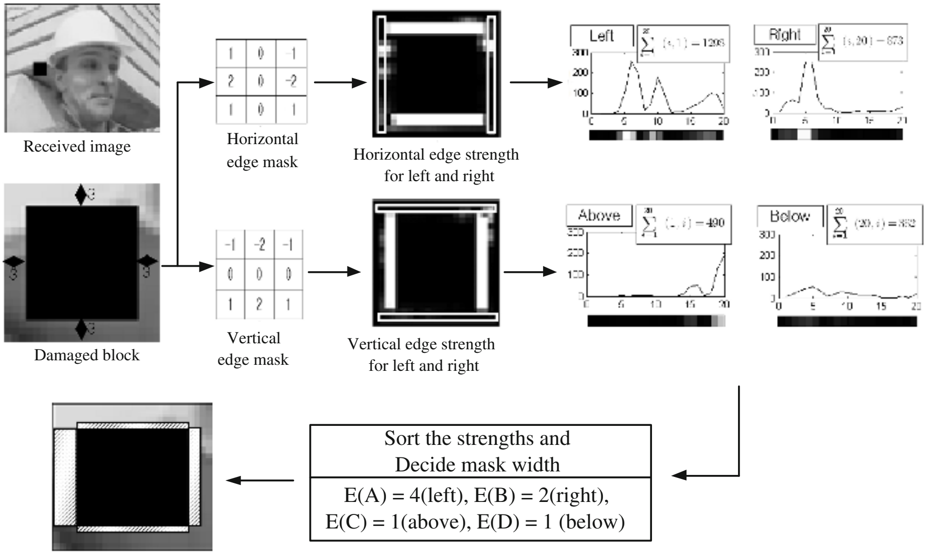


Fig. 3. Mask decision by edge information

$$D_L = \sum_{i=-1}^{N} \sum_{j=0}^{n_L-1} (\hat{f}(p+i, q-1-j) - f(p+i, q-1-j))^2 \tag{5}$$

$$D_R = \sum_{i=-1}^{N} \sum_{j=0}^{n_R-1} (\hat{f}(p+i, q+N+j) - f(p+i, q+N+j))^2 \tag{6}$$

$$D_A = \sum_{i=-1}^{N} \sum_{j=0}^{n_A-1} (\hat{f}(p-1-j, q+i) - f(p-1-j, q+i))^2 \tag{7}$$

$$D_B = \sum_{i=-1}^{N} \sum_{j=0}^{n_B-1} (\hat{f}(p+N+j, q+i) - f(p+N+j, q+i))^2 \tag{8}$$

where n_L, n_R, n_A, n_B are the widths of the masks. Since the difference in each direction is computed, we can decide the best motion vector which provides the minimum of total difference $D_{total} = D_L + D_R + D_A + D_B$. If there is the region unavailable the region is excluded in consideration. Finally, the motion vector is given by

$$MV_{rec} = arg \min_{mv} D_{total} \tag{9}$$

4 Experimental Results

To evaluate the performance of the proposed method, EA-BMA, we executed the test for QCIF Foreman, Carphone, Mobile and Stefan sequences. It is assumed

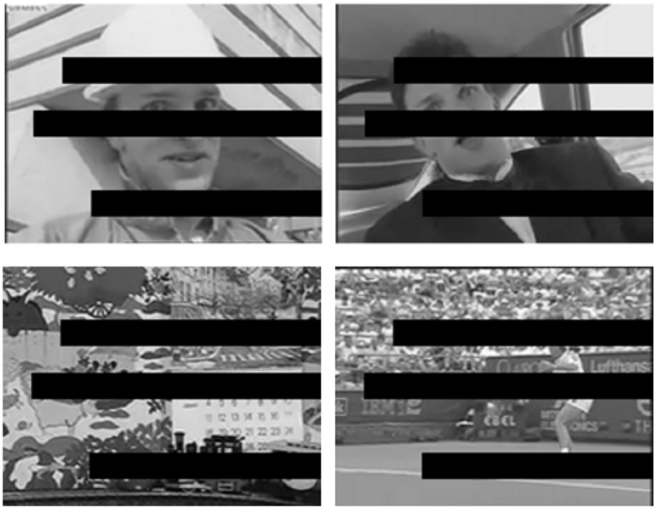


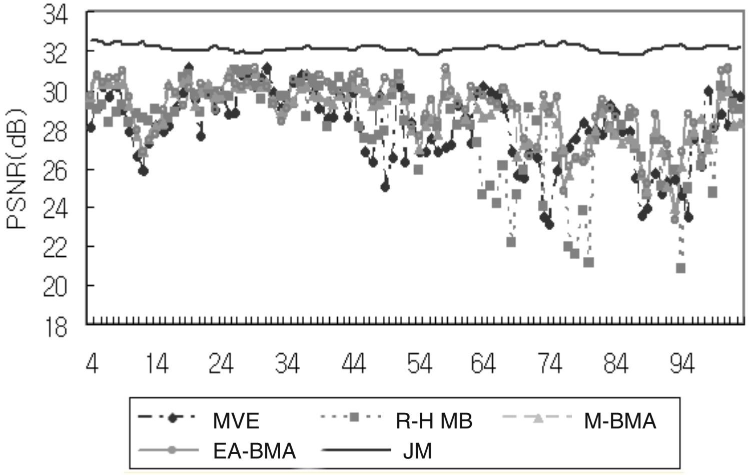
Fig. 4. Error patterns for performance evaluation

Table 1. Error concealment performance for Foreman, Carphone, Mobile, and Stefan sequences

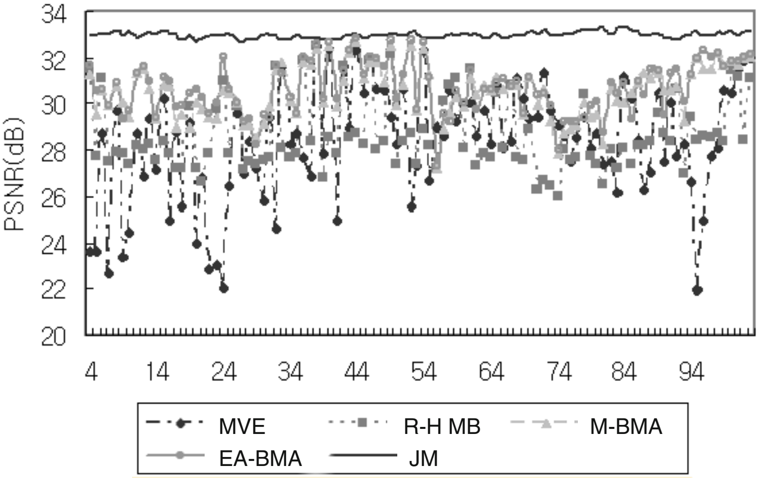
image	method (average PSNR)				
sequence	no loss	MVE	R-H MB	M-BMA	EA-BMA
Foreman	32.11	27.42	27.95	28.75	29.10
Carphone	32.98	26.76	28.59	30.51	30.71
Mobile	29.04	25.35	24.36	24.44	25.29
Stefan	30.08	24.56	24.42	23.73	25.74

that a slice consists of a single row and no error occurs in the slices above and below the slice which includes damaged blocks. This assumption can be reasonable by using appropriate error resilience techniques such as interleaving. Fig. 4 shows error patterns for test and H.264 video codec (JM 7.0) was used. The proposed method is compared with motion vector extrapolation (MVE) technique, recursive half macroblock (R-Half MB) technique, and modified BM (M-BMA) technique.

As illustrated in Table 1, in which the figures are the average PSNRs of 100 frames, the proposed method is the best in PSNR for Foreman, Carphone, and Stefan sequences, and it also gives a good performance for Mobile sequence even though MVE is the best for the sequence. The proposed method works well for Foreman and Carphone sequences that include relatively large objects, since it is motivated by the connectivity with neighboring pixels. It is because in case where an object covers the damaged region as well as the neighboring regions the connectivity can be preserved very well rather than in case where the regions include small objects. In addition, our method also works quite well even for



(a) Foreman



(b) Carphone

Fig. 5. PSNR versus frame (frame 4 to 103)

Mobile and Stefan sequence that includes many small objects. In case of Mobile sequence, our method is following just behind MVE that gives best performance. The details of the experimental results are shown in Fig. 5 where JM denotes the case of no error.

5 Conclusion

This paper presented a new temporal error concealment technique using the edge connectivity to neighboring region with a damaged region. We proposed the mask adaptive to the strength of edge in the region adjacent to the damaged block. The mask is located outside the damaged block. In matching process, we used the masks outside a damage block instead of prediction to boundary pixels of the damaged block. The best motion vector resulted from the process was used in error concealment. The experimental results showed that our method is very effective for image sequences, especially image sequences including large objects.

Acknowledgement

This work was supported by the IT Research Center (ITRC), Ministry of Information and Communication, Korea (IITA-2006-(C1090-0603-0002)).

References

1. M.E.Al-Mualla, C.N.Cangarajah and D.R.Bull, “Motion field interpolation for temporal error concealment”, IEE Proceedings-Vision Image and Signal Processing, vol.147, Issue 5, pp. 445-453, Oct. 2000.
2. W.M.Lam,A.R. Reibman, and B. Liu, “Recovery of lost or erroneously received motion vectors”, Proceedings of ICASSP, vol.5, pp.417-420, April 1993.
3. M. Ghanbari and V. Seferidis, “Cell-loss concealment in ATM video codecs,” IEEE Trans. on Circuits and Systems for Video Technology, vol. 3, no. 3, pp. 248-247, June 1993.
4. Y. Wang and Q. F. Zhu, “Signal loss recovery in DCT-based image and video codec,” Proceedings of Visual Communication and Image Processing, pp. 667-678, Nov. 1991.
5. Jian Feng, Kwok-Tung Lo and Hanssna Mehrpour, “Error concealment for MPEG video transmissions”, IEEE Trans. on Consumer Electronics, vol.43, no.2, pp.183-187, May 1997.
6. Yong H.Jung, Yong-goo Kim, and Yoonsik Choe, “Robust Error Concealment algorithm using iterative weighted boundary matching criterion”, Proceedings of International Conference on Image Processing, vol.3, pp.384-387, Sept.2000.
7. Qiang Peng, Tianwu Yang, Changqian Zhu, “Block-Based Temporal Error Concealment for Video Packet Using Motion Vector Extrapolation”, Proceedings of International Conference on Communications, Circuits and Systems and West Sino Expositions, vol.1, pp. 10-14, July 2002.
8. Mei-Juan Chen, Che-Shing Chen, Ming-Chieh Chi, “Recursive Block-Matching Principle for Error Concealment Algorithm, Proceedings of the 2003 International Symposium on Circuits and Systems, vol.2, pp. 528-532, May 2003.

Fractional Full-Search Motion Estimation VLSI Architecture for H.264/AVC

Chien-Min Ou[1], Huang-Chun Roan[2], and Wen-Jyi Hwang[2,*]

[1] Department of Electronics Engineering, Ching-Yun University,
Chungli, 320, Taiwan
cmou@cyu.edu.tw

[2] Graduate Institute of Computer Science and Information Engineering,
National Taiwan Normal University, Taipei, 117, Taiwan
g93470056.whwang@csie.ntnu.edu.tw

Abstract. A novel half-pel full-search motion estimation VLSI architecture for H.264/AVC video encoders is presented in this paper. Based on the processing element arrays eliminating redundant data accesses and attaining 100 % utilization, the architecture can be implemented with low clock rate while having high processing throughput. Such an implementation is particularly suited to applications requiring real time operations with high compression efficiency and low power.

Keywords: VLSI architecture, Video coding, Fractional motion estimation, H.264 standard.

1 Introduction

Motion estimation (ME) [4] and compensation based on block-matching operations have been extensively used for removing temporal redundancy in many video coding applications because of their simplicity and effectiveness. Although the integer pel ME may provide satisfactory reconstruction results, the fractional pel ME is necessary in many applications because of the increasing demand on high video compression quality. One conventional approach to realizing the fractional ME in hardware is a direct extension to the those for the VLSI implementation of integer pel ME, where the same processing element (PE) array for the integer pel ME is also used for the fractional pel ME [5]. In this approach, the interpolated samples in the search region will first be computed, and stored in a memory buffer for subsequent accesses. This may introduce large storage overhead. In the PE array, the search for the candidate block having minimum sum of absolute difference (SAD) should cover the candidate blocks formed by integer position and interpolated samples. Accordingly, as compared with the integer pel ME, the fractional pel ME based on this architecture has longer latency for identifying the optimal candidate block. Although the long latency can be compensated by increasing the clock rate and the source voltage of the circuit, the average power may also be increased.

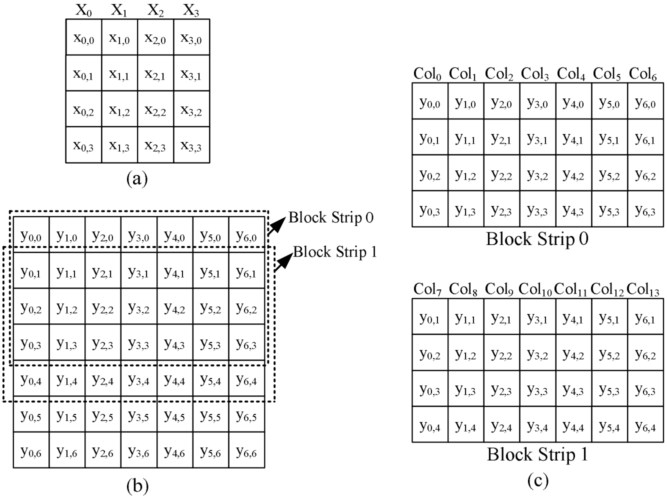
[*] Corresponding author.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 861 – 868, 2006.

The objective of this paper is to present a novel VLSI architecture for half-pel ME of H.264/AVC [4,6] having the advantages of low storage overhead, low latency and low power. In the architecture, low latency and low power are attained by the employment of concurrent PE arrays and the reduction of source voltage and clock rate. That is, we lower the power consumption and compensate the delay by increasing the silicon area [1]. There are four PE arrays in the architecture. These arrays are responsible respectively for the SAD computation of candidate blocks formed by integer position samples, vertically interpolated half-pel samples, horizontally interpolated half-pel samples, and diagonally interpolated half-pel samples. Each PE array is able to eliminate redundant accesses among adjacent candidate blocks so that it is not necessary to store the interpolated samples in the memory buffer. The storage overhead for fractional ME therefore can be minimized. The proposed architecture has been prototyped, simulated and synthesized for 0.18 μm CMOS technology using UMC standard cells. The measured data demonstrate that the architecture can be an effective alternative for the applications where high video compression quality, high computational speed and low power are desired.

2 Background

This section reviews some background material of this paper. We start with the integer-pel ME. Figure 1 shows a $N \times N$ current block $(N = 4)$ and its search area for the integer pel full-search BMA. The range of displacement is $[-p,-(p-1)]$ $(p = 2)$ in both x- and y- directions. Therefore, the size of the search region is given by $(N+2p-1)\times(N+2p-1)$. There are $2p\times 2p$ candidate blocks in the search area. The candidate blocks in the same row form a block strip. Adjacent block strips are overlapping. For the illustration purpose, the columns of the block strips are indexed as shown in Figure 1.



X_0	X_1	X_2	X_3
$x_{0,0}$	$x_{1,0}$	$x_{2,0}$	$x_{3,0}$
$x_{0,1}$	$x_{1,1}$	$x_{2,1}$	$x_{3,1}$
$x_{0,2}$	$x_{1,2}$	$x_{2,2}$	$x_{3,2}$
$x_{0,3}$	$x_{1,3}$	$x_{2,3}$	$x_{3,3}$

(a)

$y_{0,0}$	$y_{1,0}$	$y_{2,0}$	$y_{3,0}$	$y_{4,0}$	$y_{5,0}$	$y_{6,0}$
$y_{0,1}$	$y_{1,1}$	$y_{2,1}$	$y_{3,1}$	$y_{4,1}$	$y_{5,1}$	$y_{6,1}$
$y_{0,2}$	$y_{1,2}$	$y_{2,2}$	$y_{3,2}$	$y_{4,2}$	$y_{5,2}$	$y_{6,2}$
$y_{0,3}$	$y_{1,3}$	$y_{2,3}$	$y_{3,3}$	$y_{4,3}$	$y_{5,3}$	$y_{6,3}$
$y_{0,4}$	$y_{1,4}$	$y_{2,4}$	$y_{3,4}$	$y_{4,4}$	$y_{5,4}$	$y_{6,4}$
$y_{0,5}$	$y_{1,5}$	$y_{2,5}$	$y_{3,5}$	$y_{4,5}$	$y_{5,5}$	$y_{6,5}$
$y_{0,6}$	$y_{1,6}$	$y_{2,6}$	$y_{3,6}$	$y_{4,6}$	$y_{5,6}$	$y_{6,6}$

(b)

Col_0	Col_1	Col_2	Col_3	Col_4	Col_5	Col_6
$y_{0,0}$	$y_{1,0}$	$y_{2,0}$	$y_{3,0}$	$y_{4,0}$	$y_{5,0}$	$y_{6,0}$
$y_{0,1}$	$y_{1,1}$	$y_{2,1}$	$y_{3,1}$	$y_{4,1}$	$y_{5,1}$	$y_{6,1}$
$y_{0,2}$	$y_{1,2}$	$y_{2,2}$	$y_{3,2}$	$y_{4,2}$	$y_{5,2}$	$y_{6,2}$
$y_{0,3}$	$y_{1,3}$	$y_{2,3}$	$y_{3,3}$	$y_{4,3}$	$y_{5,3}$	$y_{6,3}$

Block Strip 0

Col_7	Col_8	Col_9	Col_{10}	Col_{11}	Col_{12}	Col_{13}
$y_{0,1}$	$y_{1,1}$	$y_{2,1}$	$y_{3,1}$	$y_{4,1}$	$y_{5,1}$	$y_{6,1}$
$y_{0,2}$	$y_{1,2}$	$y_{2,2}$	$y_{3,2}$	$y_{4,2}$	$y_{5,2}$	$y_{6,2}$
$y_{0,3}$	$y_{1,3}$	$y_{2,3}$	$y_{3,3}$	$y_{4,3}$	$y_{5,3}$	$y_{6,3}$
$y_{0,4}$	$y_{1,4}$	$y_{2,4}$	$y_{3,4}$	$y_{4,4}$	$y_{5,4}$	$y_{6,4}$

Block Strip 1

(c)

Fig. 1. The 4×4 current block and its search area. (a) the 4×4 current block , (b) the search area , (c) the index of columns of the first two block strips.

The 1D systolic array [2] of the full-search BMA is shown in Figure 2, which skews each column of the current blocks and candidate blocks for the SAD computation. Table 1 shows data flow schedule indicating the starting clock for calculating each column SAD, which will take N clock cycles to complete. Every N consecutive column SADs will then be accumulated as one block SAD.

3 The Proposed Architecture

Although the 1D systolic array is simple to construct, the columns of search area will be re-fetched as shown in Figure 2 and Table 1. Therefore, it is necessary to use memory buffers for data reuse.

The proposed architecture is based on the PE array presented in our previous work [3], which is used for the integer pel full-search ME for an $N \times N$ current block without the redundant accesses of candidate blocks within the same block strip. The circuit contains N 1D systolic arrays, and each systolic array contains N PEs, as shown in Figure 3.

Table 1. The data flow schedule of 1D array (N=4)

Clock Cycle	Inputs		Operations
	Current Block	Search Area	
0	X_0	Col_0	$\lvert X_0 - Col_0 \rvert$
1	X_1	Col_1	$\lvert X_1 - Col_1 \rvert$
2	X_2	Col_2	$\lvert X_2 - Col_2 \rvert$
3	X_3	Col_3	$\lvert X_3 - Col_3 \rvert$
4	X_0	Col_1	$\lvert X_0 - Col_1 \rvert$
5	X_1	Col_2	$\lvert X_1 - Col_2 \rvert$
6	X_2	Col_3	$\lvert X_2 - Col_3 \rvert$
7	X_3	Col_4	$\lvert X_3 - Col_4 \rvert$

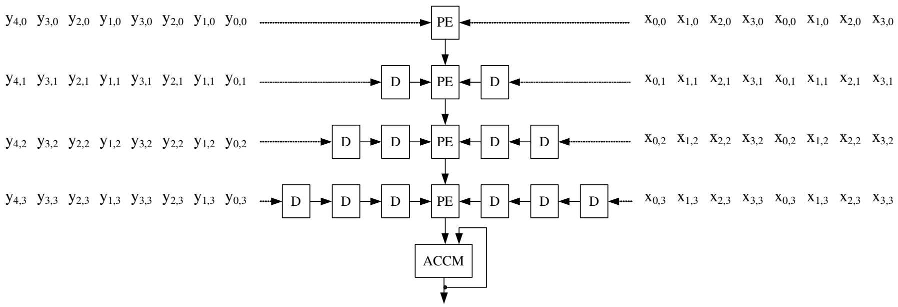

y4,0 y3,0 y2,0 y1,0 y3,0 y2,0 y1,0 y0,0
y4,1 y3,1 y2,1 y1,1 y3,1 y2,1 y1,1 y0,1
y4,2 y3,2 y2,2 y1,2 y3,2 y2,2 y1,2 y0,2
y4,3 y3,3 y2,3 y1,3 y3,3 y2,3 y1,3 y0,3
x0,0 x1,0 x2,0 x3,0 x0,0 x1,0 x2,0 x3,0
x0,1 x1,1 x2,1 x3,1 x0,1 x1,1 x2,1 x3,1
x0,2 x1,2 x2,2 x3,2 x0,2 x1,2 x2,2 x3,2
x0,3 x1,3 x2,3 x3,3 x0,3 x1,3 x2,3 x3,3
PE
D
ACCM

Fig. 2. The Basic structure of 1D array

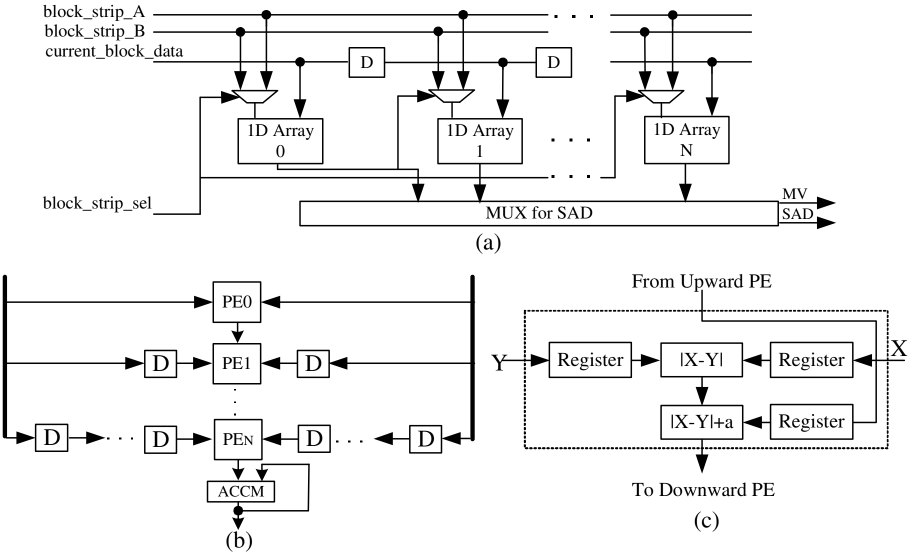


Fig. 3. The VLSI architecture of the PE array for integer pel full-search ME without redundant accesses of candidate blocks within the same block strip: (a) The architecture (b) 1D array architecture (c) PE architecture

The circuit operates by scheduling the columns of the current block through a delay line, and broadcasting columns of two adjacent candidate block strips in the search region on each clock cycle. Each 1D systolic array then skews the pixels in the input columns for SAD computations.

Table 2 shows the data flow of the PE array for the current block and its search area with N=4. In addition to having high throughput [3], the major advantage of the circuit is that each column in the same block strip is accessed only once. The redundant accesses within each block strip can then be removed. This advantage is very helpful for the half-pel ME.

In addition, from Table 2, it can be observed that the PE array produces one block SAD for each clock cycle. Define the latency of the structure as the total number of clock cycles required for identifying the candidate block having minimum SAD for each current block. Accordingly, the latency of the PE array is $2p \times 2p$. The latency of conventional 1D array shown in Figure 2 is $N \times 2p \times (2p+N-1)$ [2]. The proposed architecture therefore has lower latency over the basic 1D array.

In the H.264 half-pel ME, each half-pel sample that is horizontally or vertically adjacent to two integer samples is interpolated from integer-position samples using 6-tap finite impulse response (FIR) filter [4]. The remaining half-pel samples (termed diagonal half-pel samples) are then calculated by interpolating between six horizontal or vertical half-pel samples. Figure 4 shows the proposed architecture for the realizing the H.264 half-pel ME. It contains the FIR filter bank, and the PE arrays. The FIR filter bank calculates the horizontal, vertical and diagonal half-pel samples from the integer-position samples. The four PE arrays are then responsible for the subsequent SAD computation. All the PE arrays have identical architecture shown in Figure 3.

Table 2. The data flow of the proposed PE array shown in figure 3 for *p*=2

Clock	Current _block_ data	Block_strip_A	Block_strip_B	1-D Array 0	1-D Array 1	1-D Array 2	1-D Array 3
0	X_0	Col_0		$\lvert X_0 - \mathrm{Col}_0\rvert$			
1	X_1	Col_1		$\lvert X_1 - \mathrm{Col}_1\rvert$	$\lvert X_0 - \mathrm{Col}_1\rvert$		
2	X_2	Col_2		$\lvert X_2 - \mathrm{Col}_2\rvert$	$\lvert X_1 - \mathrm{Col}_2\rvert$	$\lvert X_0 - \mathrm{Col}_2\rvert$	
3	X_3	Col_3		$\lvert X_3 - \mathrm{Col}_3\rvert$	$\lvert X_2 - \mathrm{Col}_3\rvert$	$\lvert X_1 - \mathrm{Col}_3\rvert$	$\lvert X_0 - \mathrm{Col}_3\rvert$
4		Col_4	Col_7	$\lvert X_0 - \mathrm{Col}_7\rvert$	$\lvert X_3 - \mathrm{Col}_4\rvert$	$\lvert X_2 - \mathrm{Col}_4\rvert$	$\lvert X_1 - \mathrm{Col}_4\rvert$
5		Col_5	Col_8	$\lvert X_1 - \mathrm{Col}_8\rvert$	$\lvert X_0 - \mathrm{Col}_8\rvert$	$\lvert X_3 - \mathrm{Col}_5\rvert$	$\lvert X_2 - \mathrm{Col}_5\rvert$
6		Col_6	Col_9	$\lvert X_2 - \mathrm{Col}_9\rvert$	$\lvert X_1 - \mathrm{Col}_9\rvert$	$\lvert X_0 - \mathrm{Col}_9\rvert$	$\lvert X_3 - \mathrm{Col}_6\rvert$
7			Col_{10}	$\lvert X_3 - \mathrm{Col}_{10}\rvert$	$\lvert X_2 - \mathrm{Col}_{10}\rvert$	$\lvert X_1 - \mathrm{Col}_{10}\rvert$	$\lvert X_0 - \mathrm{Col}_{10}\rvert$
8		Col_{14}	Col_{11}	$\lvert X_0 - \mathrm{Col}_{14}\rvert$	$\lvert X_3 - \mathrm{Col}_{11}\rvert$	$\lvert X_2 - \mathrm{Col}_{11}\rvert$	$\lvert X_1 - \mathrm{Col}_{11}\rvert$
9		Col_{15}	Col_{12}	$\lvert X_1 - \mathrm{Col}_{15}\rvert$	$\lvert X_0 - \mathrm{Col}_{15}\rvert$	$\lvert X_3 - \mathrm{Col}_{12}\rvert$	$\lvert X_2 - \mathrm{Col}_{12}\rvert$
10		Col_{16}	Col_{13}	$\lvert X_2 - \mathrm{Col}_{16}\rvert$	$\lvert X_1 - \mathrm{Col}_{16}\rvert$	$\lvert X_0 - \mathrm{Col}_{16}\rvert$	$\lvert X_3 - \mathrm{Col}_{13}\rvert$
11		Col_{17}		$\lvert X_3 - \mathrm{Col}_{17}\rvert$	$\lvert X_2 - \mathrm{Col}_{17}\rvert$	$\lvert X_1 - \mathrm{Col}_{17}\rvert$	$\lvert X_0 - \mathrm{Col}_{17}\rvert$
…	…	…	…	…	…	…	…

To compute the horizontal, vertical and diagonal half-pel samples, *two adjacent candidate block strips* of integer-position samples (denoted by Block_strip_A and Block_strip_B in Figure 4) are accessed in the manner similar to the schedule shown in Table 2. Each source block strip is interpolated to form block strips of horizontal, vertical and diagonal half-pel samples by the filter bank. Since two adjacent source block strips are accessed concurrently, and each one is used to calculate three interpolated strips, the filter bank produces six interpolated block strips. It consists of six filters: F_j, j=1,..,6 (depicted in Figure 4.(b)). Each filter produce one column of the interpolated samples at a time for the subsequent PE array. The filters F_1 and F_2 operate on each single column of the input strips for generating the vertically interpolated strips on every clock cycle. The filters F_3 and F_4 perform the averaging operations across columns of the input strips for calculating the horizontally interpolated strips. Finally, the filters F_5 and F_6 are used to produce the diagonally interpolated strips by averaging pels on each single column of horizontally interpolated strips.

All the filter outputs are delivered directly to the PE arrays. It is not necessary to store the interpolated pels, because the PE arrays require no redundant access within a block strip. This can effectively reduce the storage overhead and power consumption. In addition, since four PE arrays operate concurrently, the SAD computation of integer-position samples can be performed in parallel with those of horizontal, vertical and diagonal half-pel samples.

Accordingly, the latency of the proposed architecture is $2p \times 2p$, which is identical to that of the architecture in Figure 3 for integer-pel ME. The architecture therefore attains low latency for fractional ME with low power consumption and low storage size overhead.

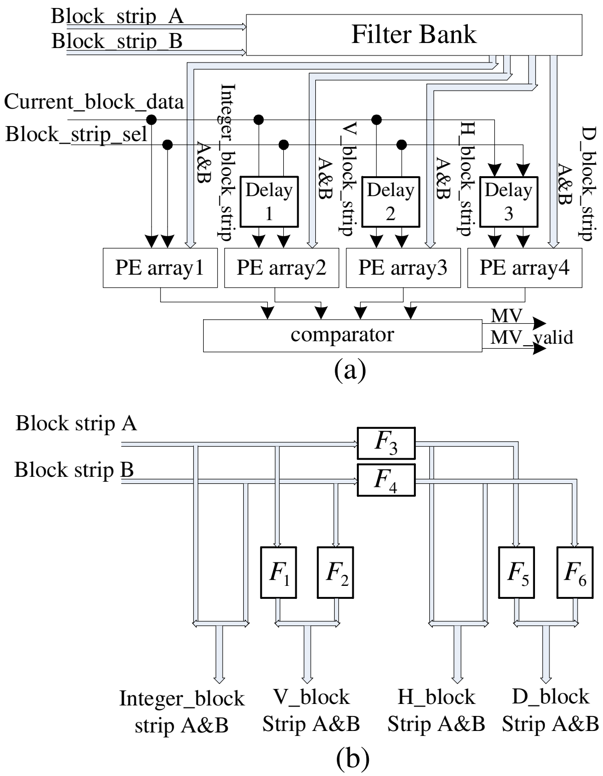


Fig. 4. Proposed half-pel ME architecture. (a) The architecture, (b) The architecture of filter bank.

Table 3. The proposed fractional me chip performance

Number of PE	4×8×8
Range of displacement	[8,-7]
Block size	8×8
Technology	UMC 0.18 μm
Gate count	351K
Max frequency	334 MHZ

4 The Experimental Results

The H.264 ME chip based on the proposed architecture was designed with Synopsys synthesis tools using a standard cell library based on the UMC 0.18 µm CMOS technology process. The main characteristics of the circuit are presented in Table 3. The circuit contains 4 × 8 × 8 PEs (N=8), with a range of displacement [8,-7] (p=8). The latency and maximum frequency of the circuit are given by 256 (i.e., $4p^2$) and 334 MHz, respectively.

Table 4 shows the required clock rates and the corresponding average power dissipation of the proposed architecture for various frame sizes and frame rates. It can be observed from the table that the clock rate for processing the high definition TV (HDTV) video sequences at 60 frames/sec is only 238 MHz, which is still lower than the maximum clock rate of the circuit as shown in Table 3.This chip therefore supports a wide range of video formats for H.264 based applications.

One major advantage of high throughput is that the clock rate is substantially reduced subject to a constraint on frame size and frame rate. The clock rate reduction may lower the power consumption [1] for ME, and is useful for low power applications.

To justify the employment of 4 PE arrays in our architecture, the clock rates and the power dissipation of the architecture containing only one PE array (termed single PE system) for half-pel ME are also included in the Table 4. In the single PE system, the integer block strip, and horizontally, vertically and diagonally interpolated strips are processed one at a time. In addition, the system should contain local RAM for storing the interpolated samples. Consequently, as shown Table 4, our architecture has lower power dissipation than the single PE system subject to the same video format. For example, the power consumption of our architecture is only 679.51 mW for HDTV sequences. On the contrary, the power consumption of single PE system is 884.72 mW for the same sequence. All these facts demonstrate the effectiveness of the proposed architecture.

Table 4. Comparisons of clock pates and average power consumption

Frame Size		CIF	4CIF	16CIF	SDTV	HDTV
Frame Rate (fps)		30	30	30	30	60
Proposed Circuit	Clock Rate (MHz)	12.16	48.64	194.64	110.59	221.18
	Power (mW)	33.23	135.95	453.15	271.91	679.51
Single-PE	Clock Rate (MHz)	48.64	194.64	778.56	442.36	884.72
	Power (mW)	66.58	251.93	887.72	531.55	1329.20

5 Conclusion

As compared with the single PE architecture for fractional ME, the proposed architecture is able to produce the best MVs with lower clock rate. Therefore, the circuit is well suited for low power designs. In particular, the clock rate for the VBS-BMA operations over CIF sequences at 30 fps is 12.16 MHz. The resulting power dissipation is only 33.23 mW, which may be attractive for mobile or portable video applications. On the other hand, the frame size and frame rate supported by the circuit can also be substantially extended subject to a clock rate constraint. In our experiment, the required clock rate for HDTV sequences at 60 fps is 221.18 MHz. The circuit may therefore be very helpful for designs requiring high visual quality. Gate-level synthesization and verification illustrate that our circuit is beneficial for enhancing the performance of H.264 encoders over a wide range of video applications.

References

1. Chandrakasan, A.P. and Brodersen, R.W. Minimizing power consumption in digital CMOS circuits, Proceedings of the IEEE, Vol. 83. (1995) 498-523
2. Kuhn, P. Algorithms, complexity analysis and VLSI architectures for MPEG-4 motion estimation, Kluwer Academic (1999)
3. Ou, C.M. Lee, C.F. and Hwang, W.J. An efficient VLSI architecture for H.264 variable block size motion estimation, IEEE Trans. Consumer Electronics, Vol. 51. (2005) 1291-1299
4. Richardson, I.E.G. H.264 and MPEG-4 Video Compression, John Wiley & Sons (2003)
5. Sayed, M. and Badawy, W. A half-pel motion estimation architecture for MPEG-4 applications, Proc. IEEE Int. Symp. on Circuits and Systems, Vol.2. (2003) 792-795
6. Wiegand, T. Sullivan, G.J. Bjontegaard, G. and Luthra, A. Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits and Systems for Video Technology, vol. 13. (2003) 560-576

Motion JPEG2000 Coding Scheme Based on Human Visual System for Digital Cinema

Jong Han Kim, Sang Beom Kim, and Chee Sun Won*

Dept. of Electronics Eng., Dongguk University,
Seoul, 100-715, Korea
{yhande, supernova, cswon}@donnguk.edu

Abstract. In this paper, the properties of human visual system such as insensitivity to high temporal frequency and contrast masking are exploited to improve the subjective coding efficiency of the Motion JPEG2000. Specifically, we detect motion information using DWT (Discrete Wavelet Transform) coefficients of the JPEG2000 and then determine the importance of the corresponding high frequency components based on contrast masking effect. If the motion turns out to be imperceptible by the human vision system, then we discard the corresponding subband data. Proposed scheme yields higher subjective quality at lower bit-rate than the conventional Motion JPEG2000 coding scheme. Also, the proposed algorithm needs lower memory access and complexity than the previous enhanced coding scheme using motion estimation.

Keywords: Motion JPEG2000, Human visual system, Rate control, Discrete wavelet transform.

1 Introduction

Motion JPEG2000 [1] is an international standard for video compression documented as JPEG2000 Part 3. Because of its high intra frame video compression efficiency, in 2004, DCI (Digital Cinema Initiatives) has selected Motion JPEG2000 as the compression format to be used for future digital distribution of motion pictures [2]. In Motion JPEG2000, each frame is coded independently by using JPEG2000 image coding scheme with a constant bit-rate [3], thus the encoded bitstream contains interframe redundancy. To reduce the redundancy, enhanced Motion JPEG2000 coding scheme has been presented [4], which adaptively controls the bit-rate for each frame by using the property of the human vision system. In the previous scheme [4], motion vectors are detected between adjacent frames by motion estimation, then an appropriate number of bits is allocated to each code-block according to the amount of the motion. The scheme can enhance the subjective video quality, but the video quality strongly depends on manually determined parameters such as the threshold of the motion vector and the weight factor of subbands. Also the previous scheme requires high computational complexity due to the estimation of the motion vectors. The goal of this paper is to further improve the efficiency of the Motion JPEG2000 in

* Corresponding author.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 869 – 877, 2006.

terms of the complexity and the subjective visual quality. To achieve this goal, we exploit human visual system, which has the insensitivity to temporal frequency as well as contrast masking effect.

It has long been studied about human visual system to be used for image compressions. There are two main properties in human visual system, namely contrast sensitivity and contrast masking [5][6], which have been used to compress images with high subjective quality. Contrast sensitivity is used to reduce the redundancy of high frequency components, where human vision system cannot perceive well both in spatial and temporal domains. For example, motion compensation is used to reduce the temporal redundancy in DCT-based and wavelet-based coding [7][8]. Another human visual property, contrast masking, has also been used to remove the redundant information in images with multiple object regions [9][10]. They exploit the contrast masking to reduce the bit-rate without visual artifacts.

In this paper, we propose a novel coding scheme, which exploits both the properties mentioned above. Since JPEG2000 is based on DWT, we can detect moving areas by calculating the difference of two low frequency subband images for the subsequent frames. Then, we exploit the contrast masking property to determine the importance of the high frequency components in the moving area. This, in turn, is used to control the bit-rate. We can improve the subjective compression efficiency by removing the temporal redundancy in the detected moving areas.

The rest of this paper is organized as follows. In section 2, the conventional JPEG2000 coding method is introduced. With a brief description of contrast masking effect, its implementation in wavelet domain is explained in section 3. In section 4, the proposed compression scheme is introduced. Simulation results are provided in section 5 followed by the conclusions in section 6.

2 JPEG2000 Coding Scheme

Conventional JPEG2000 coding scheme is shown in Figure 1. As a preprocessing, input image is partitioned into rectangular and non-overlapping spatial regions on a regular grid called tiles. Since each image consists of multiple components, tiling is applied to each component separately but in a spatially consistent way. The 2-D discrete wavelet transform (DWT) decomposes each color component into sub-images with different resolutions corresponding to the various frequency bands (LL, HL, LH, HH). Figure 2 shows the hierarchical structure of wavelet decomposition. The low-frequency sub-image (LL_0 in Figure 2) is the approximation of the original image at a coarse resolution and it maintains contextual information of the original image. High frequency sub-images (HL, LH, HH) represent vertical, horizontal, and diagonal detail information, respectively. Most of the coefficients in the high frequency band are around zero at smooth area in the original image, but take large amplitude only along the detail areas such as edges and contours of the image. DWT coefficients of each subband at different resolution are divided into codeblocks, which are the basis blocks of entropy coding with a regular size. Each quantized codeblock is coded independently by bit-plane coding (BPC) in Tier-1 coding process.

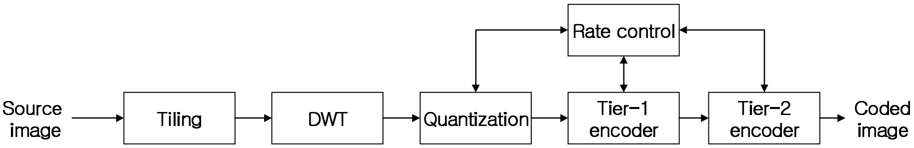


Fig. 1. Block diagram of JPEG2000 encoder

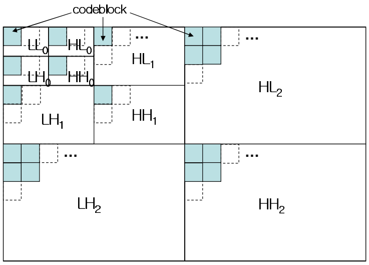


Fig. 2. Hierarchical structure of DWT

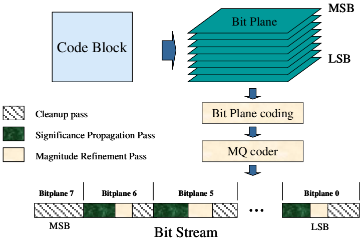


Fig. 3. Codestream organization

In BPC, bit-planes are coded by three coding passes (significant propagation, refinement and clean up) from the most significant bit to the least significant one. After the BPC, MQ-coder (which is context based adaptive arithmetic coder) generates a set of packets, which is a part of the codestream constructed by header information and the compressed image. Figure 3 represents bit-planes and codestream organization. In Tier-2 coding, rate-distortion optimization is performed. Coding passes, which come from MQ-coder, are truncated for all codeblocks by rate-distortion slope value. More information regarding this issue can be found in [3][11].

3 Human Visual System

Human visual system has the following two properties. The first is that human eye is less sensitive to higher frequency components than to lower ones and this fact has been used in the design of video equipments. The second is that the perception of human vision system will be changed when two signals (the background and the foreground) are coupled. The former is known as contrast sensitivity and the latter is contrast masking.

The contrast masking can be described such that the human visual sensitivity to gray level increment shows a nonlinear relationship with the background gray contrast [9]. For example, when the background has a gray value of c_1 and the foreground has gray value of c_2 on it, human observer can discriminate the foreground from the background only when their contrast is greater than a detection threshold. Figure 4 shows a contrast masking function diagram [5].

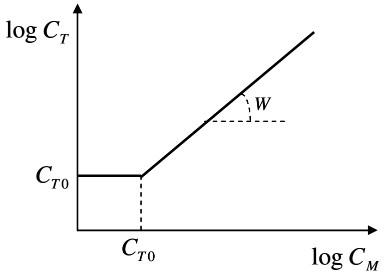


Fig. 4. A contrast masking function diagram

In Figure 4, C_{T0} denotes the detection threshold of the signal measured in the absence of masker. As one can see in the figure, if the contrast of the masker, C_M , is lower than C_{T0} , the detection threshold, C_T , remains constant (i.e., $C_T = C_{T0}$). On the other hand, when C_M is greater than C_{T0} , C_T is determined as a power of the contrast masker. In fact, the actual detection threshold C_T can be modeled as follows [5][9],

$$C_T = C_{T0} \cdot Max\left[1, \left(\frac{C_M}{C_{T0}}\right)^W\right] \tag{1}$$

where W is the slope of the masking function as illustrated in Figure 4.

One way to combine the wavelet property with the contrast masking effect was proposed by Hai and Shen [9]. The outline of their proposed method is as follows. In wavelet domain, the low-frequency sub-image (LL_0 in Figure 2) is the approximation of the original image at a coarse resolution. And three high frequency sub-images (HL, LH, HH) contain the detail information such as edges and contours with their own orientation. As a coarse approximation of the original image, the low frequency sub-image can be regarded as background contrast of the image, and the corresponding high-frequency sub-image can be regarded as the increment of gray value on the background. According to the contrast masking effect, when the high frequency coefficients are less than a certain detection threshold, they can be ignored because human vision cannot detect them. Since human visual system has a logarithmic characteristic, for a gray level g , let us define G as follows [9]:

$$G = \ln g \, . \tag{2}$$

Then, G can be used as the actual gray level perceived by human vision system. Now, the derivative of (2) yields:

$$\Delta G = \frac{\Delta g}{g}. \tag{3}$$

The increment ΔG of gray level in the original image can be replaced by the relative increment $\Delta g / g$. This means that relative increment can be used to determine the actual gray level difference of the human vision system. Similarly, the frequency difference ΔW perceived by the human vision system can be represented by the relative frequency difference $\Delta w / w$. Hai and Shen [9] use the energy of high frequency coefficients to represent Δw and corresponding low-frequency coefficients to the background contrast w, which yields the following relationship,

$$\Delta W = \frac{\Delta w}{w} \approx \frac{\| V_x^s \|^2}{\| A_x \|^2} \tag{4}$$

where x represents the corresponding resolution level and s is the frequency index such as HL, LH, HH. A and V represent vector set of low and high frequencies, respectively. In other words, A_x represents low frequency vector set at resolution level x, and V_x^s represents vector set of s th frequency at resolution level x. T_x is the detection threshold which can be determined by contrast masking function. If the ratio of the norm of the s^{th} frequency (i.e., $\| V_x^s \|^2$) and the norm of the low frequency (i.e., $\| A_x \|^2$) are less than T_x (i.e., $\left\| V_x^s \right\|^2 / \left\| A_x \right\|^2 < T_x$), then V_x^s can be discarded for the data compression, because they can not be perceived by the human vision system.

4 Proposed Coding Scheme

Figure 5 illustrates block diagram of the proposed coding scheme. The lower shaded part in the diagram contains newly added processes and the upper part is the conventional JPEG2000 coding procedure. As shown in Figure 5, only previous low frequency subband coefficients are needed in the proposed scheme for the motion detection.

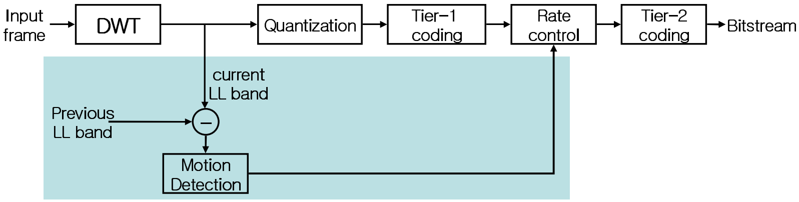


Fig. 5. Block diagram of the proposed scheme

For low computational complexity, we use a temporal differencing method to detect motion between two consecutive frames. The perceptible motion can be defined to be the noticeable difference between the two frames. To determine the just noticeable difference, we employ the mapping function proposed by Ward [12]. When the energy difference, $D_{i,j}$, of the $(i,j)^{\text{th}}$ macro blocks between the n^{th} and $(n-1)^{\text{th}}$ frames (the energies are denoted respectively as $E_{i,j}^{n-1}$ and $E_{i,j}^{n}$) is larger than the threshold $\Delta E\left(E_{i,j}^{n}\right)$

$$D_{i,j} = \left|E_{i,j}^{n-1} - E_{i,j}^{n}\right| < \Delta E(E_{i,j}^{n})\,. \tag{5}$$

Then, it is regarded as a motion block. The threshold $\Delta E\left(E_{i,j}^{n}\right)$ can be determined as follows [12],

$$\Delta E(E_{i,j}^{n}) = 0.0594 \cdot (1.219 + E_{i,j}^{n}{}^{0.4})^{2.5} \tag{6}$$

The motion detection is based on the macro blocks in the low frequency sub-image, which corresponds to codeblocks of the JPEG2000 in the high frequency sub-image at the highest resolution. Figure 6-(a) illustrates the motion detection process. White areas are macro blocks with detected motion. Since each codeblock is coded independently, detected macro blocks are collected to a codeblock. Figure 6-(b) shows collected codeblocks at resolution level 2. These codeblocks are determined to have motion and their coefficients are the candidates for discarding.

The amount of the data reduction allocated to a codeblock is determined by considering the importance of codeblocks. Important codeblocks are the ones with high frequency contrast. Thus the coefficients in the important codeblocks are coded with the conventional R-D optimization while those in unimportant codeblocks are discarded. Note that the frequency contrast can be measured by (4).

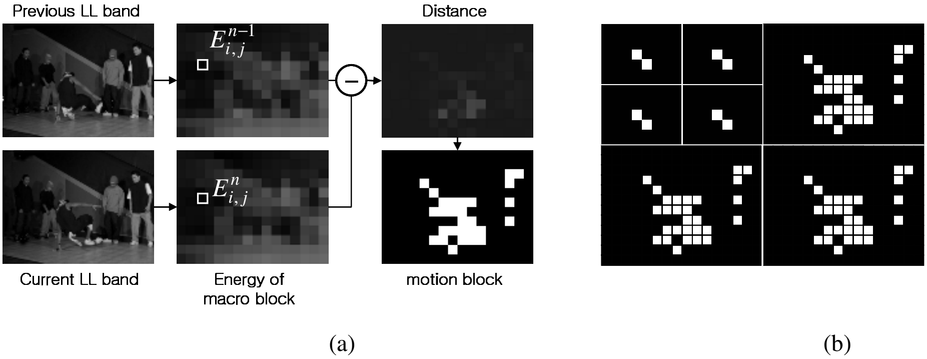


Fig. 6. Motion detection: (a) motion detection process, (b) collected codeblocks with motion blocks for each subband

Figure 7 shows the flowchart of the proposed scheme. In Figure 7, R_x^s is the ratio of the norm of the high frequency and the norm of the low frequency subbands (i.e., $R_x^s = \left\| V_x^s \right\|^2 / \left\| A_x \right\|^2$) described in section 3. Since we use the property of human visual system, the additional data reduction can be performed without noticeable distortion.

When the abrupt shot change occurs, the number of detected motion blocks becomes larger. In such a case, in the proposed method, it is possible that the unexpected distortion increases. Thus in the proposed scheme, it is regarded as a shot change when the number of detected motion blocks is greater than 60% of the total number of motion blocks in the image. And, in such a case, our coding method is not applied.

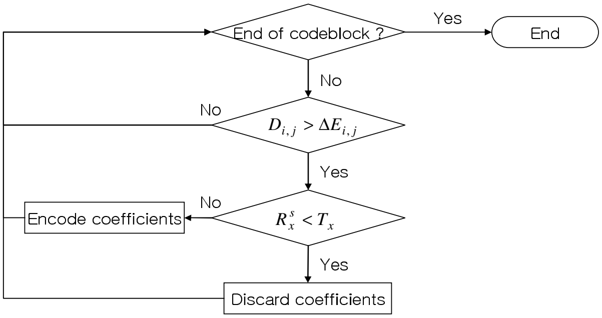


Fig. 7. Flowchart of the proposed scheme

5 Simulation Results

This section describes the coding results at various bit-rates for test video streams. Test video streams used in our experiments are provided by the video quality experts group (VQEG)[13]. Japer-1.701.0 is used for implementation of the proposed scheme and encoding options are from the DCI specification [3]. A video stream is encoded by the proposed scheme at various bit-rates and then assessed by 15 reagents using double-stimulus impairment scale method with conventional reference streams. We follow the test conditions in [14]. The test results are indicated by MOS (Mean Opinion Score) with 5 levels. Here, level 5 means the highest subjective quality, and level 1 means the lowest one. To assess the subjective quality, we compare the MOS of the proposed method with the conventional MJPEG2000 (i.e., Jasper 1.701.0). Table 1 shows the MOS results for various bit-rates. As one can see, the performance of the proposed method is better than the conventional one. In this experiment, we achieved approximately 10% improvement of the bit-rate than the conventional one at the same MOS.

The removal of the high frequency components introduces the decrease of objective quality measure, PSNR. The PSNR decrement of the proposed method is about 0.7dB. Note that the additional distortion occurs only when the motion is detected and the energy of the corresponding high frequency component is low enough. In other words, the additional distortion may not be noticeable because human cannot perceive it.

Table 1. Experiment result in MOS (Mean Opinion Score)

Video size	Proposed method		Conventional MJPEG2000	
	bit-rate	MOS	bit-rate	MOS
720×480	0.10	4.0	0.12	3.5
	0.19	4.2	0.21	4.2
	0.27	5.0	0.30	4.8
1024×768	0.10	4.1	0.12	4.0
	0.19	4.4	0.21	4.4
	0.27	4.6	0.30	4.6

In the proposed scheme, we need additional processing time of only 0.4% of the motion estimation using the full search, which is small enough to be ignored in JPEG2000 encoding time.

6 Conclusion

In this paper, we propose a novel Motion JPEG2000 coding scheme based on human visual system. Exploiting the insensitivity of the human vision system to high temporal frequency and the frequency contrast masking, we can achieve additional compression in the wavelet subband areas. Specifically, we discard the coefficients which human observer cannot detect artifacts. Thus, the proposed scheme enhances subjective quality with a lower bit-rate. Our scheme needs previous low frequency subband coefficients only, thus the additional computational complexity is negligible.

Acknowledgement

This work was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul Industry-Academy-Research Cooperation Project and the MIC, Korea, under the ITRC support program supervised by the IITA (IITA-2005-(C1090-0502-0027)).

References

1. ISO/IEC 15444-3 : 2002(E), Information technology - JPEG2000 image coding system - Part 3 : Motion JPEG2000, 2002.
2. DCI Digital Cinema System Specification V1.0, http://www.dcimovies.com
3. ISO/IEC 15444-1 : 2004(E), Information technology - JPEG2000 image coding system : Core coding system
4. Miyamoto. R, Sugita. H, Hayashi. Y, Tsutsui.H, Masuzaki.T, Takao.O and Nakamura.Y, "High quality Motion JPEG2000 coding scheme based on the human visual system", Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on 23-26 May 2005 Page(s):2096 - 2099 Vol. 3
5. Andrew B,Watson, "Efficiency of a Model Human Image Code", Journal of Optical Society of America, Vol.4,pp.2401-2417,1987

6. G.E.Legge and J.M.Foley, "Contrast masking in human vision", Journal of Optical Society of America, Vol.70,pp.1458-1471, 1980
7. J. R. Ohm, "Three-dimensional subband coding with motion compensation," IEEE Trans.Image Process, Vol.3, no.5, pp.559-589, Sep.1994
8. S. Choi and J. Woods, "Motion-compensated 3-D subband coding of video", IEEE Trans. Image Process, Vol.8, no.2, pp.155-167, Feb.1999
9. Hai Wei, Lansun Shen, "Fractal Coding of Wavelet Image Based on Human Vision's Contrast Masking Effect", Proceedings of SPIE - Volume 3959, Human Vision and Electronic Imaging V, Bernice E. Rogowitz, Thrasyvoulos N. Pappas, Editors, June 2000, pp. 584-594
10. Etoh. M, Kobayashi. M, Adachi. S, "Closed loop optimization of image coding using subjective error criteria", Acoustics, Speech, and Signal Processing.2001. Proceedings.(ICASSP'01).2001 IEEE international conference on Vol.3, pp 1717-1720, 7-11. May. 2001
11. D. S. Taubman and M. W. Marcellin, JPEG2000: Image Compression Fundamentals, Standards, and Practice, Kluwer Academic, 2002.
12. G. Ward, ``A Contrast-Based Scalefactor for Luminance Display'', Graphics Gems IV, Ed. by P. S. Heckbert, pp. 415-421, 1994.
13. http://www.vqeg.org
14. CCIR-500, Method for the subjective assessment of the quality of television pictures, CCIR Recommendation 500-3, 1986.

Wavelet-Based Image Compression with Polygon-Shaped Region of Interest

Yao-Tien Chen[1], Din-Chang Tseng[1], and Pao-Chi Chang[2]

[1] Department of Computer Science and Information Engineering,
National Central University, Chung-li, Taiwan
{ytchen, tsengdc}@ip.csie.ncu.edu.tw
[2] Department of Communication Engineering, National Central University, Chung-li, Taiwan
pcchang@ce.ncu.edu.tw

Abstract. A wavelet-based lossy-to-lossless image compression technique with polygon-shaped *ROI* function is proposed. Firstly, split and mergence algorithms are proposed to separate concave *ROIs* into smaller *convex ROIs*. Secondly, row-order scan and an adaptive arithmetic coding are used to encode the pixels in *ROIs*. Thirdly, a lifting integer wavelet transform is used to decompose the original image in which the pixels in the *ROIs* have been replaced by zeros. Fourthly, a wavelet-based compression scheme with adaptive prediction method (*WCAP*) is used to obtain predicted coefficients for difference encoding. Finally, the adaptive arithmetic coding is also adopted to encode the differences between the original and corresponding predicted coefficients. The proposed method only needs less shape information to record the shape of *ROI* and provides a lossy-to-lossless coding function; thus the approach is suitable for achieving the variety of *ROI* requirements including polygon-shaped *ROI* and multiple *ROIs*. Experimental results show that the proposed lossy-to-lossless coding with *ROI* function reduces bit rate as comparing with the *MAXSHIFT* method in *JPEG2000*; moreover, when the image without *ROI* is compressed by the proposed lossless coding, the proposed approach can also achieve a high compression ratio.

Keywords: Image compression, region of interest (*ROI*), lossy-to-lossless coding, *ROI* coding, difference encoding.

1 Introduction

Image compression is used to reduce the image data size as small as possible under a tolerance limit of errors. In general, the techniques of image compression can be classified into two major categories: loss and lossless. Lossy compression requires not only less storage space, but also less transmission time or bandwidth, while lossless compression can completely reconstruct the original data. In addition to offering high-quality compression, an effective approach to image compression should further incorporate value-adding functions, such as *ROI* coding and lossy-to-lossless coding. A *ROI* refers to a special region in an image that is of particular interest or imperative importance to the user who can free to identify the *ROI* based on ones needs. In

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 878 – 887, 2006.

general, an image can be separated into important and non-important parts for a particular purpose; the important part represented by the *ROI* is often compressed by a lossless style while the non-important part can be compressed by a lossy-to-lossless style to achieve a tradeoff between the fidelity and the coding efficiency. That is, the *ROI* undergoes lossless compression first and then finer details of the remaining part of the image are gradually added at a later stage to achieve lossy-to-lossless coding.

A lot of different techniques of *ROI* coding have been proposed recently. Li and Li [1] proposed shape adaptive discrete wavelet transform (*SA-DWT*) for arbitrarily shaped object coding. With the use of the transform, the spatial correlation and wavelet transform properties, such as locality property and self-similarity across subbands, are preserved. Tasdoken and Cuhadar [2] proposed region-based integer wavelet transform (*RB-IWT*) as an alternative to *SA-DWT*. The *RB-IWT* enables lossless coding of image regions which can not be achieved by *SA-DWT* due to the fixed-precision representation of wavelet coefficients. Fukuma *et al.* [3] introduced a switching wavelet transform by using shorter-length basis for *ROI* and longer-basis for non-*ROI*. The bases with different lengths provide better compression quality than a fixed-length wavelet transforms. Liu *et al.* [4] proposed a method for chromosome image compression which combines lossless compression of chromosome *ROIs* with lossy-to-lossless coding for the remaining image parts. The method performs a differential operation on chromosome *ROIs* for decorrelation, and is followed by integer wavelet transforms on *ROIs* and the remaining image parts. The boundary of chromosome *ROIs* are then traced by chain code method.

The model of *ROI* coding supported in *JPEG2000* is based on scaling the wavelet coefficients. The technique can be further classified into two different methods: general scaling-based and *MAXSHIFT* method [5]. For a general scaling-based method, a shape encoder/decoder is required to encode/decode the shape information (*i.e.*, the shape of *ROI*). This makes both encoder and decoder more complicated and increases the bit rate; moreover, the method needs a *ROI* mask indicating which wavelet coefficients have to be transmitted exactly in order for the receiver to reconstruct the desired region perfectly. In contrast, the *MAXSHIFT* method does not need the shape information. However, if there are multiple *ROIs* with different degrees of interest, the *MAXSHIFT* method has to handle more difficult problems than a general scaling method, since the dynamic range has to be increased significantly.

To solve the mentioned problems, we propose a wavelet-based image compression technique with polygon-shaped *ROI* and lossy-to-lossless coding. Firstly, split and mergence algorithms are proposed to separate concave *ROIs* into smaller *convex ROIs*. Secondly, row-order scan and an adaptive arithmetic coding are used to encode the pixels in *ROIs*. Thirdly, a lifting integer wavelet transform is used to decompose the original image in which the pixels in the *ROIs* have been replaced by zeros. Fourthly, a wavelet-based compression scheme with adaptive prediction method (*WCAP*) is used to obtain predicted coefficients for difference encoding. Finally, the adaptive arithmetic coding is also adopted to encode the differences between the original and corresponding predicted coefficients. We only need less shape information to achieve the polygon-shaped *ROI* and multiple *ROIs* with different degrees of interest. Furthermore, the proposed approach does not need to generate the *ROI* mask.

The remaining sections of this paper are organized as follows. In Section 2, we present the proposed approaches: split and mergence algorithms, lifting integer wavelet transform, and the *WCAP* method. Experiments are reported in Section 3. Conclusions are given in Section 4.

2 The Proposed Approach

The block diagram of the proposed approach shown in Fig. 1 is composed of seven processes: *ROI* selection, *Graham*'s scan algorithm [6], split and mergence algorithms, row-order scan, lifting integer wavelet transform, *WCAP* method [7], and adaptive arithmetic coding.

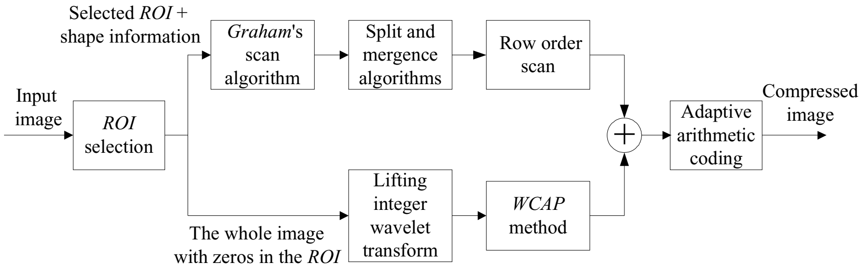


Fig. 1. The block diagram of the proposed approach

ROIs are always formed by polygons or circles. When a *ROI* is formed by a polygon, the *ROI* is represented by the *ordered vertices* of the polygon and the *ordered vertices* just constitute the shape information. A polygon R in an image is called convex if line segment $\overline{ab}$ for any pair of pixels a, b in R is completely in R. The *convex hull* of a polygon R is the smallest *convex polygon* containing R and represented by its *convex vertices*. Examples of polygon-shaped *ROI* and *convex polygon* are given in Fig. 2. In Fig. 2 (a), a polygon-shaped *ROI* is represented by *ordered vertices*: P_1, P_2, P_3, P_4, P_5, P_6, P_7, and P_8, where P_i, $1 \le i \le 8$, are denoted by coordinates in the image. In Fig. 2 (b), a *convex polygon* is represented by corresponding *convex vertices*: P_1, P_2, P_3, P_4, and P_5. On the other hand, when the *ROI* is formed by a circle, the shape of *ROI* is represented by the center point and radius of the circle. Similarly, the center point and radius just constitute the shape information.

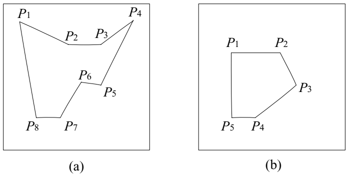


Fig. 2. Examples of polygon-shaped *ROIs* and *convex polygons*. (a) A polygon-shaped *ROI* and the *ordered vertices*. (b) *A convex polygon* and the *convex vertices*.

2.1 Split and Mergence Algorithms

Given the shape information, both the encoder and the decoder use split and mergence algorithms to split the concave *ROI* into multiple *convex ROIs*. Then we can simply use row-order scan to extract all coefficients in all shaped *ROIs*. Hence, the encoder just needs to transmit very little shape information to the decoder, and the decoder can perfectly reconstruct the *ROIs*. The split and mergence algorithms consist of three steps: (*i*) judging whether a given *ROI* is convex, (*ii*) exploiting split algorithm to separate the *ROI* into multiple non-overlapped *convex ROIs* if necessary, and (*iii*) merging two or more *ROIs* to constitute a larger *convex ROI* if the mergence still satisfies the condition of *convex hull*.

A given *ROI* is convex if each *ordered vertex* is exactly scanned once by *Graham*'s scan algorithm, and these vertices must form a *convex ROI*; otherwise, split algorithm will be used to achieve the necessary condition of *convex hull*. Split algorithm is a recursive approach dividing a *ROI* into multiple non-overlapped smaller *convex ROIs* step by step. The algorithm is divided into the following four steps:

Step 1. Identify the *convex ROI* by tracing vertex one after one in the given *ordered vertices*. If any current vertex violates the condition of *convex hull*, a smaller *convex ROI* will be successfully split. Meanwhile, the algorithm pushes a vertex prior to the current vertex into a *temporary queue*.

Step 2. The split is processing until the given *ordered vertices* is empty, and one or more *convex ROIs* are obtained upon the completion of this step.

Step 3. All vertices in the *temporality queue* are ejected to compose new *ordered vertices* for subsequent steps.

Step 4. Repeat from *Step*1 until no *ROI* to be split.

After all *non-convex ROIs* are split, mergence algorithm is then performed to merge adjacent *convex ROIs* into larger convex *ROIs* by the following two steps:

Step 1. Any two adjacent *convex ROIs* (*i.e.*, there exists a common boundary between the two *ROIs*) is merged to generate a larger *convex ROI* if they satisfy the condition of *convex hull*.

Step 2. The result of *Step* 1 is regarded as the input of mergence algorithm for the next step until no two adjacent *ROIs* can be further merged.

Illustration of the split and mergence algorithms is shown in Fig. 3. An original *ROI* composed of *ordered vertices*: P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, and P_{11} is given in Fig. 3 (a). The first splitting *convex ROI* (P_2, P_3, P_4) is shown in Fig. 3 (b); then four *convex ROIs*, (P_2, P_3, P_4), (P_4, P_5, P_6), (P_6, P_7, P_8), and (P_8, P_9, P_{10}, P_{11}), are successively split as shown in Fig. 3 (c). Meanwhile, the *ordered vertices* (P_1, P_2, P_4, P_6, P_8, P_{11}) is pushed into the *temporary queue* and regarded as an input *ROI* in the second step as shown in Fig. 3 (d). Two *convex ROIs*, (P_1, P_2, P_4) and (P_4, P_6, P_8, P_{11}), are split as shown in Fig. 3 (e). The *ordered vertices* (P_1, P_4, P_{11}) forms a *convex ROI* in this step; thus the split algorithm stops and mergence algorithm starts as shown in Fig. 3 (f). Two *convex ROIs*, (P_1, P_2, P_4) and (P_1, P_4, P_{11}), are merged to constitute (P_1, P_2, P_4, P_{11}) as shown in Fig. 3 (g). Two *convex ROIs*, (P_4, P_5, P_6) and (P_4, P_6, P_8, P_{11}), are merged to constitute (P_4, P_5, P_6, P_8, P_{11}) as shown in Fig. 3 (h).

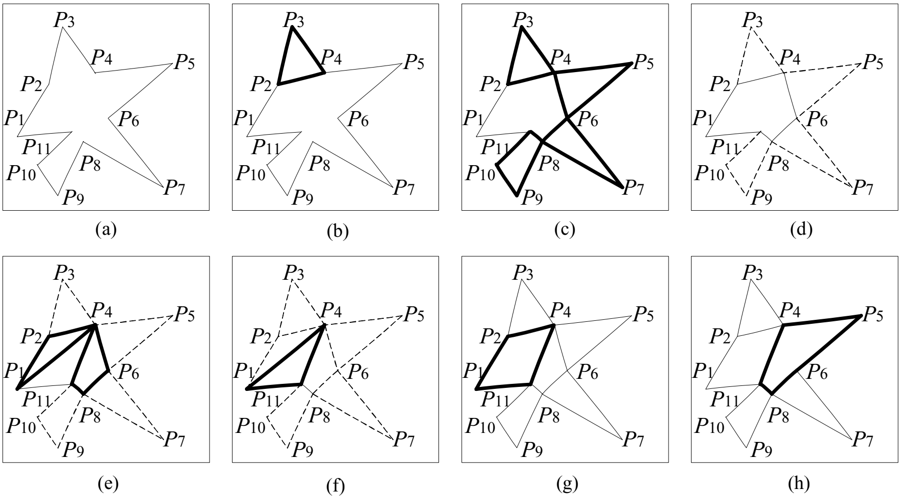


Fig. 3. Illustration of the split and mergence algorithms. (a) Original *ROI*. (b) The first *convex ROI* is split. (c) Four *convex ROIs* are obtained after the first step. (d) (P_1, P_2, P_4, P_6, P_8, P_{11}) forms an input *ROI* in the second step. (e) Two *convex ROIs*, (P_1, P_2, P_4) and (P_4, P_6, P_8, P_{11}), are obtained after the second step. (f) (P_1, P_4, P_{11}) forms a *convex ROI*. (g) (P_1, P_2, P_4) and (P_1, P_4, P_{11}) are merged to generate (P_1, P_2, P_4, P_{11}). (h) (P_4, P_5, P_6) and (P_4, P_6, P_8, P_{11}) are merged to constitute (P_4, P_5, P_6, P_8, P_{11}).

For each non-overlapped *convex ROI*, *convex vertices* are linked to form the boundary of the *convex ROI* and then row-order scan is adopted to scan all coefficients in the multiple *convex ROIs*. Thus we can precisely identify all coefficients in the non-overlapped *convex ROI*. For circular-shaped *ROI*, shape information is directly used to identify all coefficients in the *ROI* without using split and mergence algorithms. Thus we demonstrates that the proposed approach indeed only needs less shape information to achieve the *ROI* coding as compared with the conventional shape coding methods.

The advantage of the *MAXSHIFT* method surpassed to the general scaling-based method is that the *ROI* coding does not need shape information at the decoder. However, generating the *ROI* mask still remains at the encoder, and the *ROI* mask calculation is complicated. Thus the computational complexity of these two methods is relatively higher than that of split and mergence algorithms. Since the proposed approach does not need to generate the *ROI* mask and is easy to implement, it is more efficient for lossless compression with *ROI* selection.

2.2 Lifting Wavelet Transform

Our wavelet transform was implemented by lifting scheme performed with the following three steps: (*i*) *split* step for sorting the input into the even and the odd entries, (*ii*) *prediction* step for giving the value at the even entries and predicting the value at the odd entries, and (*iii*) *update* step for updating even entries up to date to reflect knowledge of the input. Lifting integer wavelet transform means that the wavelet transform can transform integers to integer coefficients and perfectly reconstruct the original integers from the integer coefficients. Lifting integer wavelet transform is capable of accomplishing fast in-place computation and especially appropriate for lossless data compression. A variety of transforms can be applied for lossless data compression. Nevertheless, according to the suggestions of the previous work [7], *S+P* transform is generally considered to be the best one. The *S+P* transform is described as

$$\begin{aligned} &d^{(1)}[n] = x[2n+1] - x[2n], \\ &s[n] = x[2n] + \left\lfloor \frac{d^{(1)}[n]}{2} \right\rfloor, \\ &\text{and} \\ &d[n] = d^{(1)}[n] + \left\lfloor \frac{2}{8}(s[n-1]-s[n]) + \frac{3}{8}(s[n]-s[n+1]) + \frac{2}{8}d^{(1)}[n+1] + \frac{1}{2} \right\rfloor, \end{aligned} \tag{1}$$

where [.] is a notation of signal, $d[n]$ and $s[n]$ are the highpass and lowpass coefficients respectively after the transform, $x[.]$ denotes the original signal, and $\lfloor . \rfloor$ is a truncation operator.

2.3 WCAP Method

The *WCAP* method was proposed by Chen *et al.* [7]. Initially, the method analyzes the higher-correlation coefficients, where wavelet coefficients are regarded as the predictor (independent) and response (dependent) variables of a prediction equation. Then based on the higher-correlation coefficients, the method launched the selection of predictor variables using a conditional statistical test to determine which relative predictor variables should be included in the prediction equation. The generated prediction equations are then applied to predict most wavelet coefficients except the lowest-resolution coefficients.

In most previous studies, the predictions were generally conducted with a fixed number of predictor variables at fixed locations. Actually, every kind of images not only has its own statistical distribution but also demonstrates different properties in different wavelet subbands. To achieve a more accurate prediction for compression, the number of predictor variables must be adaptively adjusted based on the image's properties. Thus instead of relying on a fixed number of predictors on fixed locations, the *WCAP* method uses adaptive prediction approach to overcome the *multicollinearity* problem and takes the wavelet interscale persistence and intrascale clustering properties to achieve high compression ratio.

In general, the probability distribution of the symbols to be encoded is unknown. Thus a method called adaptive arithmetic coding [8] which is combined from an adaptive probability estimation and an arithmetic coding is pursued to increase compression ratio. Adaptive arithmetic coding uses a real number to represent a sequence of symbols and updates the probability of symbol based on distribution of input symbol, whenever getting one input symbol. Finally, compression of images is achieved via this adaptive arithmetic coding.

3 Experiments

In our experiments, all test images are 512×512 gray-level images as shown in Fig. 4. The *ROIs* are manually selected. If there are *ROIs* selected, the *ROIs* will be encoded by lossless style and the remaining parts are encoded by lossy-to-lossless style; otherwise, the entire image will be encoded by lossless style. At first, the polygon-shaped *ROIs* and progressive lossy-to-lossless coding were examined to demonstrate the abilities of the proposed approach as shown in Fig. 5 (a). In Fig. 5 (b), a *ROI* was selected on the *lena*'s face and partial hat for lossless encoding. The remaining part of the image was gradually added to reconstruct the original image. As indicated from Figs. 5 (c) to (i), the remaining part was divided into eight bitplanes and starts bitplane encoding from most significant bit (*MSB*) to least significant bit (*LSB*) to achieve progressive lossy-to-lossless coding.

The comparison of bit rates among adaptive arithmetic coding, *MAXSHIFT* method, and the proposed approach is shown in Table 1. From the table, we find that the proposed approach has the best compression rate for all six standard images. To understand the improved degree of the proposed approach over other methods in compression rate, we here define an improvement ratio (*IR*) to evaluate the improvement of compression performance for method *B* over method *A* as

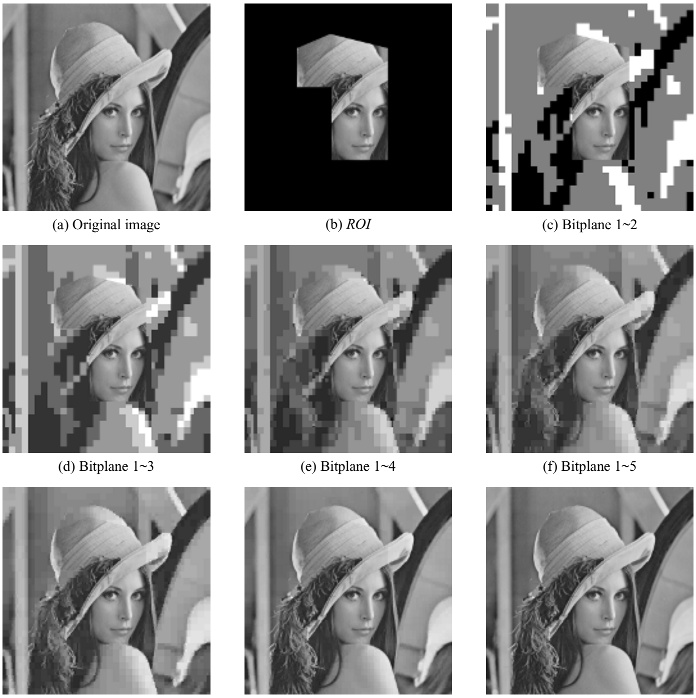
$$IR = \frac{\text{bitrate of method } A - \text{bitrate of method } B}{\text{bitrate of method } A} \times 100\% . \tag{2}$$

Here, the *MAXSHIFT* method was adopted to evaluate the improvement ratio of the proposed approach. One polygon-shaped *ROI* was selected on each of the six test images; the *ROI* approximately covers 15 - 25% of the images. The improvement ratios are given in Table 1. From the table, we find that the improvement ratios of the proposed method over the *MAXSHIFT* method are approximately 2.01 - 6.02%.

(a) *Lena* (b) *Goldhill* (c) *Boat*

(d) *Barbara* (e) *Baboon* (f) *Airplane*

Fig. 4. Six test gray-level images



(a) Original image

(b) *ROI*

(c) Bitplane 1~2

(d) Bitplane 1~3

(e) Bitplane 1~4

(f) Bitplane 1~5

(g) Bitplane 1~6

(h) Bitplane 1~7

(i) Bitplane 1~8

Fig. 5. Progressive lossy-to-lossless coding with polygon-shaped *ROI*

Table 1. The comparison of bit rates among adaptive arithmetic coding, *MAXSHIFT* method, and the proposed approach

Method / Image	*Adaptive arithmetic coding*	*MAXSHIFT method*	*The proposed approach*	Improvement ratio
Lena	4.86	4.65	4.49	3.44%
Goldhill	5.44	5.23	5.01	4.21%
Boat	5.02	4.65	4.37	6.02%
Barbara	5.71	5.19	4.95	4.62%
Baboon	6.62	6.48	6.35	2.01%
Airplane	4.62	4.04	3.81	5.69%

The compressions without *ROI* selection were also examined. The comparison of the proposed method with other lossless coding techniques: *CALIC* [9] and *JPEG2000* are given in Table 2. From the table, we can find that the proposed approach offers a better performance than the other two methods. The improvement ratios of the proposed method over the *JPEG2000* and *CALIC* methods are from 7.36 to 10.7% and from 1.69 to 5.46%, respectively.

As indicated by the above experiments, the main contributions of the proposed approach are to offer the high-performance polygon-shaped *ROI* coding and progressive lossy-to-lossless coding. Moreover, the proposed approach is also superior to *JPEG2000* and *CALIC* methods for lossless compression without *ROI* selection.

Table 2. Comparison of lossless compression for *JPEG2000*, *CALIC*, and the proposed approach in bits/pixel

Method Image	*JPEG2000*	*CALIC*	*The proposed approach*
Lena	4.33	4.10	4.01
Goldhill	4.85	4.58	4.33
Boat	4.42	4.15	4.08
Barbara	4.81	4.54	4.42
Baboon	5.98	5.66	5.54
Airplane	3.82	3.55	3.44

4 Conclusions

In this paper, a wavelet-based image compression technique with polygon-shaped *ROI* function and lossy-to-lossless coding was proposed. Firstly, split and mergence algorithms were proposed to separate concave *ROIs* into smaller *convex ROIs*. Secondly, row-order scan and an adaptive arithmetic coding were used to encode the pixels in *ROIs*. Thirdly, a lifting integer wavelet transform was used to decompose the original image in which the pixels in the *ROIs* had been replaced by zeros. Fourthly, a wavelet-based compression scheme with adaptive prediction method (*WCAP*) was used to obtain predicted coefficients for difference encoding. Finally, the adaptive arithmetic coding was also adopted to encode the differences between the original and corresponding predicted coefficients.

The proposed approach possesses the following advantages: (*i*) only needing less shape information to reconstruct the *ROIs*, (*ii*) providing a progressive lossy-to-lossless coding, achieving polygon-shaped *ROIs*, and supporting multiple *ROI*s. Thus the proposed approach is suitable for achieving the variety of *ROI* requirements. Experimental results show that the proposed lossy-to-lossless coding with *ROI* is superior to the *MAXSHIFT* method in *JPEG2000*; moreover, for lossless compression without *ROI* selection, the proposed approach has also obtained the best performance.

Now, the *ROIs* are manually selected; further study on automatic determination of *ROIs* will be achieved by integrating the level set methods [10].

References

1. Li, S., Li, W.: Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. IEEE Trans. Circuits and Systems for Video Technology, vol.10 (2000) 725-743
2. Tasdoken, S., Cuhadar, A.: ROI coding with integer wavelet transforms and unbalanced spatial orientation trees. In Proc. of the 25th Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society, vol.1 (2003) 841-844
3. Fukuma, S., Tanaka, T., Nawate, M.: Switching wavelet transform for ROI image coding. IEICE Trans. Fund. Electron. Commun. Comput. Sci., vol.E88, (2005) 1995-2005
4. Liu, Z., Xiong, Z., Wu, Q., Wang, Y., Castleman, K.: Cascaded differential and wavelet compression of chromosome images. IEEE Trans. on Biomedical Engineering, vol.49 (2002) 372-383
5. Christopoulos, C., Askelof, J., Larsson, M.: Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard. IEEE Signal Processing Letters, vol.7 (2000) 247-249
6. Berg, M. de, Kreveld, M. van, Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer, Berlin (2000)
7. Chen, Y.-T., Tseng, D.-C., Chang, P.-C.: Wavelet-based medical image compression with adaptive prediction. In Proc. of Int. Sym. on Intelligent Signal Processing and Communication Systems, Hong Kong (2005) 825-828
8. Moffat, A.,: Linear time adaptive arithmetic coding. IEEE Trans. Inform. Theory, vol.36 (1900) 401-406
9. Wu, X., Memon, N.,: Context-based lossless interband compression-extending CALIC. IEEE Trans. Image Processing, vol.9, (2000) 994-1001
10. Chan, T. F., Vese, L. A.,: Active contours without edges. IEEE Trans. Image Processing, vol.10 (2001) 266-277

Buffer Occupancy Feedback Security Control and Changing Encryption Keys to Protect MOD Services*

Jen-Chieh Lai and Chia-Hui Wang

Department of Computer Science and Information Engineering,
Ming Chuan University, Taiwan
{s3450203@ss24, wangch@}.mcu.edu.tw

Abstract. We present an effective security control for multimedia streaming in public network via buffer-occupancy feedback controls of client's playback buffer. There is a wide variety of multimedia-on-demand (MOD) applications can benefit from this end-to-end security control to protect the streaming data. Because the encryption/decryption for video packets are time-consuming process, our observation is that the playback buffer occupancy (PBO) can simply indicate the time availability to adjust the security level to affect the sending rate for the packet delivery. That is to say, adjusting security level can be applied further to keep the PBO away from buffer overflow and underflow. Therefore, we propose a feedback control of PBO not only to protect the MOD services from eavesdroppers, but also maintain the playback quality. To further boost the protection, we also apply Diffie-Hellman key negotiation method to provide the dynamic key changes while the PBO is controlled at a stable range for a long period. Moreover, due to the network uncertainty, different content delivery in MOD service will preserve different running PBOs. Then, they may have different applied security levels and key changes in service sessions. It will make eavesdroppers more difficult to recover the whole encrypted media data. In this paper, we also demonstrate the performance of encryption protection and preserved playback quality in our proposed schemes by experiments on a true VOD system with well-known encryptions of DES, 2DES and 3DES.

Keywords: MOD, encryption/decryption, playback buffer occupancy (PBO), feedback control.

1 Introduction

Based on the great absorption and acceptability of multimedia, diversified multimedia network applications such as e-learning, digital library, video on demand, video conference and video surveillance play a very important role on the prevalent Internet to affect human daily life. Now the infrastructure of broadband connections is almost furnished in the Internet, so the effective delivery of diversified media content is the key to the success for Internet business.

* The work was partially done while the author was visiting the Institute of Information Science, Academia Sinica, Taiwan in 2005 and supported by National Science Council, Project No. NSC 94-2213-E-130-001, Taiwan.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 888 – 897, 2006.

Due to the open architecture of Internet, lots of companies, research organizations and even individuals can easily work together to make their business progress through the public Internet community. But, the open architecture also makes the delivered content vulnerable to attack and eavesdrop. Computer hackers can take advantage of the ubiquitous connectivity of Internet through wired or wireless devices [8] to easily start attacking and stealing others' private data on the Internet. The multimedia data also preserve privacy and confidentiality while delivering them through the Internet. Therefore, it is important to protect the real-time media streaming applications from eavesdroppers.

Nowadays most proposed video streaming security mechanisms are strongly dependent on the coding scheme of media. So, we classify them as media-level protection scheme. One kind of the media-level protection is called partial encryption [2][3][4][5]. Either intra-frames or frame headers are encrypted through a secret key. Each intra-frame of the video streaming preserves whole image picture information. Besides, the frame header also preserves the important attributes of the related video frame. So receivers without the secrete key to decrypt the intra-frame or frame header cannot playback the partially encrypted video data.

Another kind of media-level protection scheme makes transpositions on the blocks of a singe image picture. It's hard to perceive the content in the scrambled data. We believe the complexity to recover the scrambled image without the transposition key is much the same as to solve the jigsaw puzzle.

The above-mentioned video security schemes are strongly dependent on the compression or coding schemes for different media. While providing media-level video protection scheme, applications need to spend time to parse the bit sequence of the video stream to find out the location of the intra-frames or frame header.

Most developers for real-time video streaming applications chose to ignore security provision completely because security services might content a lot of resources to reduce the playback performance for real-time video streaming application such as the MOD service. In this paper, we will propose a feedback control scheme in application-level to effectively protect the MOD service from eavesdropper without degrading the playback quality.

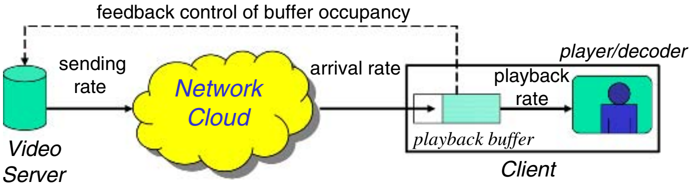


Fig. 1. MOD service session scheme

In a MOD service session as shown in Fig1, a video data requested by a user will be divided into fix-sized packets and the server will deliver them to the public network by a nominal sending rate. In the MOD client, a playback buffer is needed to accommodate data packets with varied arrival rates due to the uncertainty (i.e. delay jitter and loss) in public network. However, due to the prevalent frame-based video coding scheme and each video frame is assembled by different number of packets, the playback rate for arrival packets in playback buffer may vary. Thus, the occupancy of playback buffer (PBO) may also vary not only due to the network uncertainty of delay jitter and packet loss, but also because of client's playback rate.

However, PBO variation may incur buffer overflow and underflow to seriously degrade the playback quality. As shown in Fig1, the PBO can be controlled by applying feedback control of PBO to adjust the sending rate in MOD server to stay away from overflow and underflow. Because increasing sending rate may raise low PBO and decreasing may lower high PBO, the controlled PBO will maintain the playback quality.

Our further observation to secure MOD service is that the high PBO implies that sender has more time to apply the stronger encryption methods with more time-consuming process, and the low PBO implies that the sender should pick up easier encryption method with less time-consuming process or even aborts encryption. We believe that the dynamic encryption changes will not only strengthen the security of MOD service in public network, but also help to control PBO running at a given range.

Moreover, while the PBO is controlled within a stable range and then the playback quality is guaranteed, it is not necessary to change encryption method. However, a single key may be used for a long time while the buffer is controlled and then it is vulnerable for eavesdropper to break this key. Then the dynamic key changes can be applied to further protect the MOD service.

Therefore, our proposed security method for MOD service will apply not only the feedback control of PBO to dynamically change the applied encryption methods, but also dynamic key changes to boost the security protection. In this paper, we will first introduce the Buffer-occupancy Feedback Security Control (BFSC) and dynamic key changes in following section. The experiments and performance results to validate our proposed security schemes are investigated in section 3. The conclusion and future work are presented in the final section.

2 Buffer-Occupancy Feedback Security Control with Dynamic Key Changes

In this section, we present the buffer-occupancy feedback security control (BFSC) and dynamic key changes respectively to effectively protect the MOD service in public network. BFSC will change either the encryption method or the packet sending rate according to the PBO feedback control to maintain the running PBO. Dynamic key changes are applied while the PBO is running at a target range for a long period to strengthen the protection from eavesdropper.

2.1 PBO Feedback Security Control

The proposed BFSC scheme is simply illustrated in Fig 2. The MOD server encrypts the packets of media data at a fixed rate and then sends the encrypted packets at a nominal rate to the public network. While the encrypted packets arrive in the client, the client decrypts the arrival packets at a fixed rate and then moves decrypted packets to the playback buffer and player will playback the packets at a nominal playback rate.

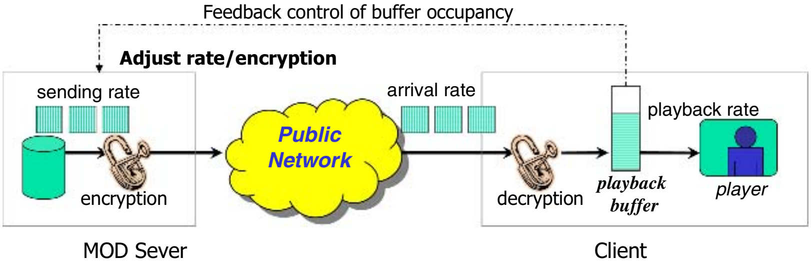


Fig. 2. Buffer-occupancy Feedback Security Control (BFSC)

Because the packets are delivered to the public network with uncertain performance, we cannot predict the delay jitter and loss for the delivered packets. The arrival rates of delivered packets won't be exactly the same as their sending rates. Furthermore, the playback rate for the packets in the playback buffer may vary because of the prevalent frame-based video coding and the video content. For the example of MPEG-n, there are three types of video frame: I-frame, P-frame and B-frame. I-frame is so-called intra-frame with the compression information of whole picture. Both P-frame and B frame are so-called inter-frame with compression information of the difference between the current and previous pictures. I-frame is assembled by much more number of fix-sized packets than P-frame and B-frame. Video content will affect the sizes of video frames and then the number of packets to assemble video frame will also vary. Then, the frame type and video content will affect the playback rate of the packets in playback buffer.

Therefore, the PBO will vary due to the variation of arrival rates and playback rates. The PBO variation should be controlled in a stable range to keep away from the buffer overflow and underflow because they will jeopardize the playback quality. A rate-based feedback control [6][9] of buffer occupancy is usually applied to control the PBO to maintain the playback quality.

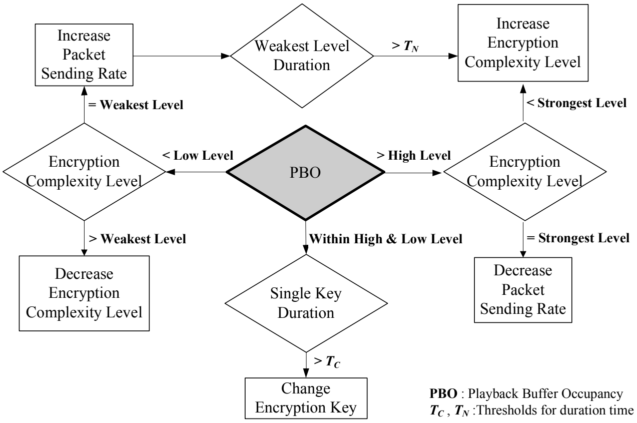

Increase Packet Sending Rate
Weakest Level Duration
> T_N
Increase Encryption Complexity Level
= Weakest Level
< Strongest Level
Encryption Complexity Level
< Low Level
PBO
> High Level
Encryption Complexity Level
> Weakest Level
Within High & Low Level
= Strongest Level
Decrease Encryption Complexity Level
Single Key Duration
Decrease Packet Sending Rate
> T_C
Change Encryption Key
PBO : Playback Buffer Occupancy
T_C , T_N :Thresholds for duration time

Fig. 3. The flow chart of BFSC with changing key

However, to further protect the MOD service, our observation is that high PBO indicates more time to apply new encryption with higher-complexity instead of decreasing sending rate to lower the PBO, and the low PBO indicates to apply new encryption with lower complexity instead of increasing sending rate to raise the PBO. The detail of BFSC flow chart is shown in Fig 3. Therefore, during the MOD service session, BFSC will dynamically change the encryption functions to enhance the strength of security protection and maintain the playback quality with the controlled running PBOs as well.

Furthermore, we must consider two other cases to provide security control for MOD service without degrading the QoS while deploying the BFSC scheme. One case is that if the PBO does not run within the high and low levels and BFSC finds no further encryption function to change, BFSC must directly increase or decrease the nominal packet sending rate according to the feedback of PBO. The other case is that once the PBO runs within the target position for a long period. We will discuss this case for further details in the following sub-section.

2.2 Dynamic Key Changes

During the MOD service session, while the PBO runs within the target levels for a long period, the encryption function will also remain unchanged for a long period to reveal vulnerability to eavesdropper. To enhance the robustness of protection for MOD streaming service in such a case, we have to change encryption keys during this long period. However, if all the keys we need to apply are stored in a database, it will cost a lot of storage space for the MOD systems, especially for the portable devices with scarce resources.

To overcome resource expense in dynamic key changes, we apply Diffie-Hellman key negotiation method to generate the first encryption key during MOD service initialization, and then we use the exponential and modular function applied in Diffie-Hellman key negotiation method as the key generation function (KGF) to generate the keys for dynamic key changes. The proposed scheme of BFSC with dynamic key changes is illustrated in Fig 4. Once the server uses the KGF to change the encryption

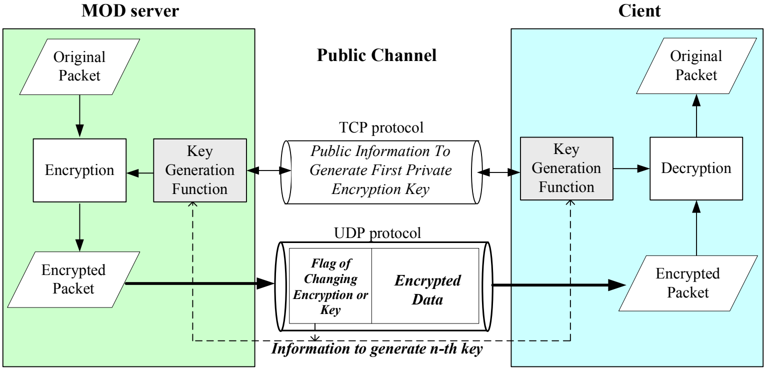


Fig. 4. Proposed scheme of BFSC with dynamic key changes

key for data packet, the flag header of changing key in the delivered packet will indicate that the encryption key has been changed to a new one. Then, the client will use the same KGF as the MOD server's to generate the same encryption key to decrypt the encrypted data.

The complete flow of dynamic key changes with Diffie-Hellman is illustrated in Fig 5. We use Diffie-Hellman key negotiation method to negotiate the first encryption key K_0 used privately in MOD server and client by exchanging information of public keys Y_s and Y_c including basis α and exponent q in the exponential function. While the server needs to change current encryption key K_{n-1} to a new encryption key K_n to boost the protection, server and client can use the same KGF to generate the new key K_n to perform the encryption and decryption respectively without troubles.

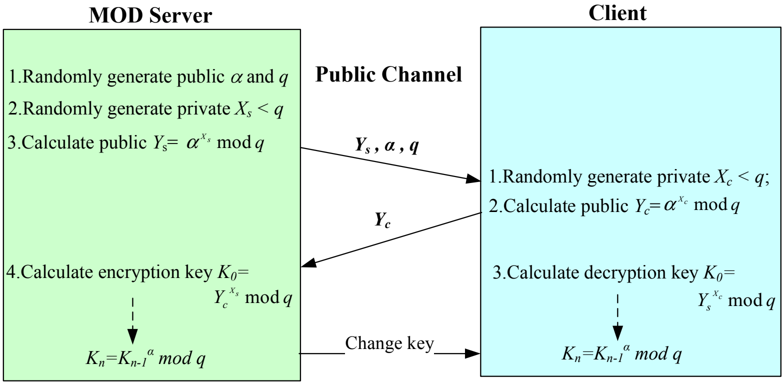

MOD Server
Client
Public Channel
1.Randomly generate public α and q
2.Randomly generate private $X_s < q$
3.Calculate public $Y_s = \alpha^{X_s} \bmod q$
Y_s , α , q
1.Randomly generate private $X_c < q$;
2.Calculate public $Y_c = \alpha^{X_c} \bmod q$
Y_c
4.Calculate encryption key $K_0 = Y_c^{X_s} \bmod q$
$K_n = K_{n-1}^{\alpha} \bmod q$
3.Calculate decryption key $K_0 = Y_s^{X_c} \bmod q$
$K_n = K_{n-1}^{\alpha} \bmod q$
Change key

Fig. 5. Dynamic key changes with Diffie-Hellman key negotiation

3 Experiments and Performance Results

In this section, we will first introduce the experiments of applying BFSC with dynamic key changes on a true VOD system. Then, we analyze the experimental results and validate the performance for the proposed security scheme.

3.1 Experiments

The test-bed for the proposed BFSC scheme is shown in Fig. 6. The experiments to demonstrate the performance of BFSC with dynamic key changes are conducted on a true VOD system by using Dummynet [10] to simulate the network uncertainty of delay, loss and bandwidth. We have 3 different types of VOD services, the first one is the No-Encryption VOD service with no encryption in data packets, the second one is the No-BFSC VOD service with a single encryption method applied during the service session and the last one is BFSC VOD service with our proposed scheme of BFSC with dynamic key changes.

In our experiments, we also examine 3 different kinds of MPEG-1 videos in our test cases, the first one is Star War-Episode I with a lot of scene changes and motions,

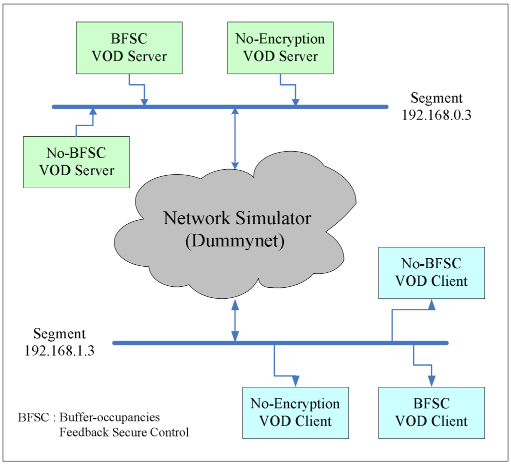


Fig. 6. Test-bed of proposed BFSC experiments

the second one is Weatherman with less scene changes and motions and the last one is football. Then, we apply 3 different kinds of encryption methods in our experiments: DES, 2DES and 3DES. The network delays inserted by Dymmynet are 0ms, 50ms, 250ms and 500ms. Moreover, the rate adjustment function Δ*r*(*t*) of buffer occupancy feedback control is shown below:

$$\Delta r(t) = (b_m - b(t))\frac{\Delta r_M}{b_m} \tag{1}$$

Δr_M is the maximum rate adjustment to avoid large burst traffic flooding to the network. $b(t)$ represents the current buffer occupancy feedback and b_m is the target position of buffer occupancy. b_m is the middle level in the playback buffer and Δr_M is as high as 220K bytes per second in our experiments. The feedback interval of buffer occupancy control is 10ms. The high level and low level to change encryption method are 60% and 40% of the playback buffer size respectively in the experiments for BFSC VOD service.

3.2 Experimental Results and Performance Analysis

In following figures, we illustrate the experimental results of PBO controllability and the sending rates to the network for No-Encryption, No-BFSC and BFSC VOD services respectively. The experiments for PBO controllability are conducted by not only inserting different network delays of 0, 50, 250 and 500ms, but also three different types of videos as mentioned before.

The Fig 7 shows the PBO controllability and sending rates to the network in No-Encryption VOD service with different inserted network delays for video Star War. The Fig 8 shows the PBO controllability and sending rates with different videos while inserted network delay is 250ms.

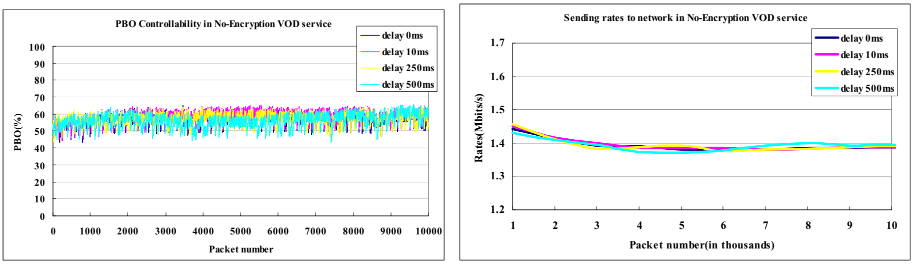


Fig. 7. PBO controllability and network sending rates in No-Encryption VOD service with different delays

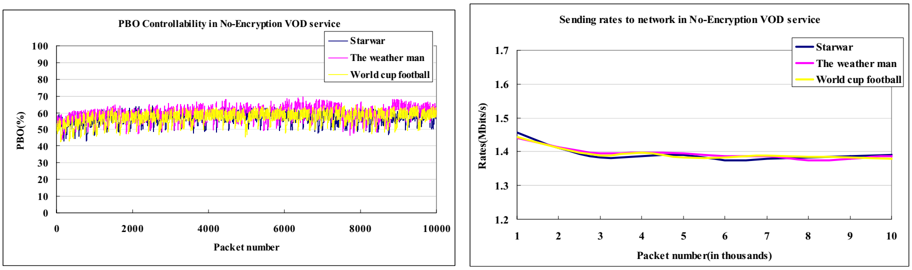


Fig. 8. PBO controllability and network sending rates in No-Encryption VOD service with different videos

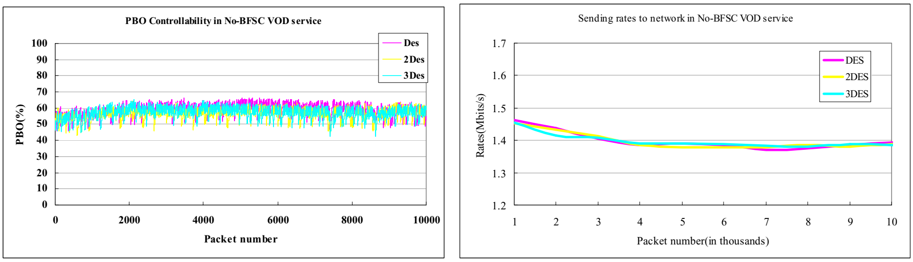


Fig. 9. PBO controllability and network sending rates in No-BFSC VOD service with different encryptions

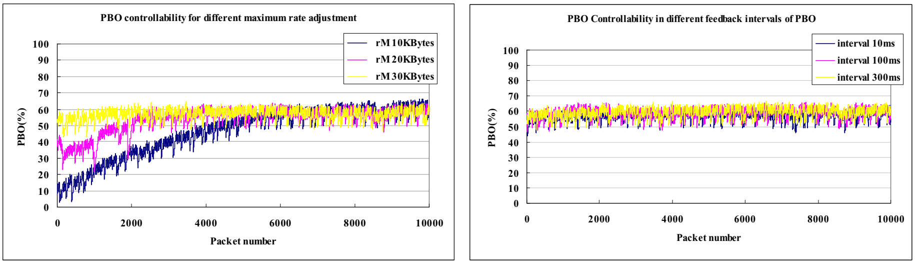


Fig. 10. PBO controllability in different maximum rate adjustment and different feedback intervals of PBO at network delay 250ms

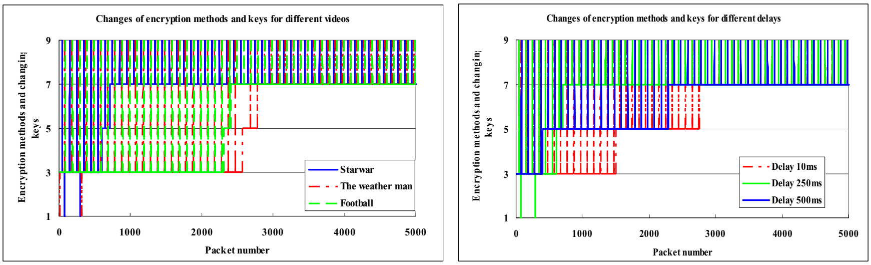


Fig. 11. Variation of encryption methods and key changes while applying different films and network delays in BFSC VOD service. (1: No encryption, 3: DES, 5: 2DES, 7: 3DES, 9: Changing Key).

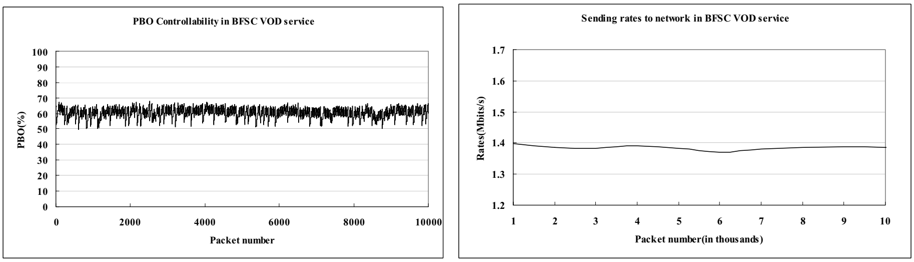


Fig. 12. PBO controllability and network sending rates in BFSC VOD service with video Star War and network delay 250ms

The Fig 9 shows the PBO controllability and sending rates to the network in No-BFSC VOD service with different encryption methods for the video Star War. While using different values of maximum rate adjustment Δr_M and different lengths of the PBO feedback interval, the PBO controllability is demonstrated in Fig 10.

The Fig 11 shows the variations of encryption methods and key changes during the BFSC VOD service session while we apply different films and network delays. The PBO controllability and network sending rates for BFSC VOD service while applying the video Star War and network delay 250ms are shown in Fig 12.

According the performance results of PBO controllability and sending rates to the network, our proposed BFSC scheme with dynamic key changes has better results than a single encryption method used in the No-BFSC VOD service.

4 Conclusions

In this paper, we propose a security scheme of BFSC with dynamic key changes for real-time MOD services by applying not only the feedback control of PBO to dynamically change the encryption methods, but also Diffie-Hellman key negotiation method to dynamically change encryption keys. Our experiments and performance results conducted on a true VOD system indicate the feasibility in our proposed security scheme without incurring much overhead to the system and network. We believe that our proposed protection scheme can provide more secure protection for MOD service in public network without degrading the playback quality.

References

1. Han-Chieh Chao, T.Y.Wu and Jiann-Liang Chen: Security-enhanced packet video with dynamic multicast throughput adjustment. International Journal of Network Management, 11(2001) 147-159
2. Yongcheng Li, Zhigang Chen, See-Mong Tan, Roy H. Campbell: Security-enhanced MPEG player. Proceedings of International Workshop on Multimedia Software Development, 25-26 March(1996)
3. Kunkelmann, T., Reinema, R.: A scalable security architecture for multimedia communication standards. Proceedings of IEEE International Conference on Multimedia Computing and Systems '97, 3-6 June (1997) 660 - 661
4. Lei Tang: Methods for encrypting and decrypting MPEG video data efficiently. Proceedings of the fourth ACM international conference on Multimedia, February (1997)
5. Tosun, A.S., Feng, W.-C.: Efficient multi-layer coding and encryption of MPEG video streams. IEEE International Conference on Multimedia and Expo, ICME 2000, Volume: 1, 30 July-2 Aug. (2000) 119 - 122
6. Chia-Hui Wang, Ray-I Chang, Jan-Ming Ho, Shun-Chin Hsu: Rate-Sensitive ARQ for Real-Time Video Streaming. GLOBECOM'03, IEEE 2003 Global Communications Conference, 1 - 5 December (2003)
7. William Stallings: Cryptography and Network Security, Principles and Practices. Prentice Hall, (2003)
8. Chui Sian Ong, Klara Nahrstedt and Wanghong: Qualily of Protection for Mobile Multimedia Applications. IEEE International Conference on Multimedia and Expo, 2003. ICME 2003. (2003) II137-II140
9. Hui Zhang, Domenico Ferrari: Rate-Controlled Service Disciplines. Journal of High Speed Networks, 3(4), (1994)
10. Luigi Rizzo: Dummynet:a simple approach to the evaluation of network protocols. Computer Communication Review, (1997) 31-41

H.264-Based Depth Map Sequence Coding Using Motion Information of Corresponding Texture Video

Han Oh and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Korea
{ohhan, hoyo}@gist.ac.kr

Abstract. Three-dimensional television systems using depth-image-based rendering techniques are attractive in recent years. In those systems, a monoscopic two-dimensional texture video and its associated depth map sequence are transmitted. In order to utilize transmission bandwidth and storage space efficiently, the depth map sequence should be compressed as well as the texture video. Among previous works for depth map sequence coding, H.264 has shown the best performance; however, it has some disadvantages of requiring long encoding time and high encoder cost. In this paper, we propose a new coding structure for depth map coding with H.264 so as to reduce encoding time significantly while maintaining high compression efficiency. Instead of estimating motion vectors directly in the depth map, we generate candidate motion modes by exploiting motion information of the corresponding texture video. Experimental results show that the proposed algorithm reduces the complexity to 60% of the previous scheme that encodes two sequences separately and coding performance is also improved up to 1dB at low bit rates.

Keywords: Three-dimensional television, depth-image-based rendering, depth map sequence coding, H.264/AVC video coding standard.

1 Introduction

We have a highly advanced visual sense that can perceive stereoscopic effects and the depth of an object. Three-dimensional television (3D-TV) is one of the promising next-generation multimedia appliances which exploits this visual sense and provides viewers with more realistic and immersive impression. In recent years, considerable research projects have been conducted on 3D-TV. Among various research efforts, a noticeable research project for 3D-TV is the Advanced Three-dimensional Television System Technologies (ATTEST) project [1] that started in March 2002. In ATTEST, a two-dimensional video sequence and its associated per-pixel depth information are recorded and transmitted, instead of transmitting two monoscopic video sequences for the left and right eye viewpoints. Stereoscopic videos are then generated using a depth-image-based rendering (DIBR) technique, as shown in Fig. 1. This data format has some advantages of high coding efficiency and interactivity as well as supporting scalability for various types of receivers.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 898 – 907, 2006.

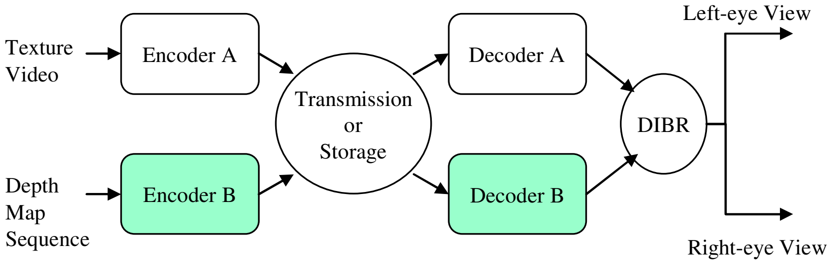


Fig. 1. Structure of 3D-TV for DIBR

In order to utilize limited bandwidth or storage space efficiently, the texture video can be compressed by exploiting its temporal and spatial correlations. In the same manner, the depth map sequence also has a significant amount of redundancies that can be removed for compression. Depth map coding schemes can be categorized into two classes. One class of depth map coding contains coding schemes using adaptive 3D mesh-based interpolation and node tracking. The other class includes coding schemes using conventional video standards, such as MPEG-2, MPEG-4, and H.264.

Mesh-based approaches [2, 3] are based on non-uniform image sampling. While the number of sampling points is reduced in flat areas, more points are coded in high curvature areas. The position and the number of points are determined by an iterative scheme based on the mean-squared-error (MSE) criterion. In order to reduce temporal redundancy, they handle inter frames by moving some nodes from frame to frame and coding the corresponding vectors. Thus, only those changes between two frames are coded. This approach is programmed with the OpenGL library and handled by a graphics hardware; therefore, its rendering time is very fast.

Another approach for depth map coding employs the conventional video coding standards [4]. 8-bit quantized depth values are mapped to a YUV color signal where UV values are set to 128, and compressed as the texture video. This scheme is easy to adapt and very efficient. Up to now, H.264 provides superior compression results compared to the mesh-based approach in terms of PSNR values [5].

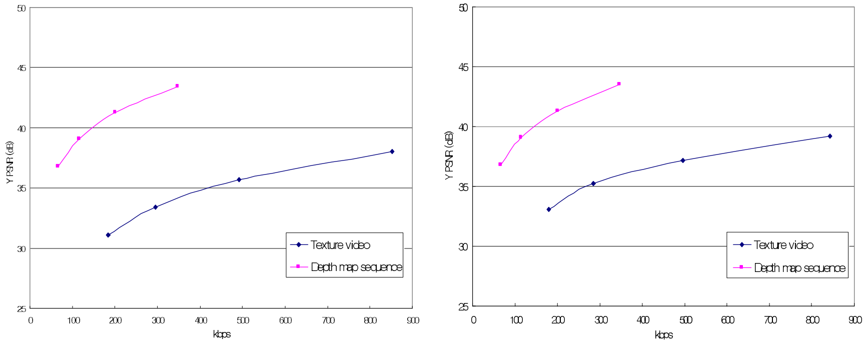
Although H.264 supports various block sizes and rate-distortion optimization and it shows high compression efficiency in depth map coding, it has some disadvantages of requiring long encoding time and high encoder complexity. This drawback is mainly due to the motion estimation operation that is performed for both sequences.

In this paper, we propose an efficient algorithm for depth map coding by sharing the motion information with the corresponding texture video. Since motion estimation in the depth map sequence is skipped and motion information is taken from the texture video, the entire encoding cost is significantly reduced. The idea of sharing motion information can be found in Stefan's paper [6]; however, it is based on MPEG-2 and motion information of the corresponding texture video is merely copied without any additional modifications. Therefore, this approach does not work well for H.264 since the motion information of the depth map sequence is not optimized as in the corresponding texture video. In this paper, we propose a new idea for sharing motion information between two sequences in H.264.

This paper is organized as follows. Section 2 briefly explains the characteristics of depth map sequence coding in H.264. Section 3 analyzes the similarity of motion information between the depth map sequence and the corresponding texture video. After we describe details of the proposed algorithm in Section 4, experimental results are presented in Section 5. Finally, Section 6 concludes this paper.

2 Characteristics of Depth Map Sequence Coding in H.264

While the texture image indicates intensity values of each color component, the depth map represents depth information per pixel. Pixels in one object tend to have similar values in both the texture image and the depth map. Since the depth map is simpler than the corresponding texture image, we can have higher coding efficiency in depth map coding, as shown in Fig. 2.



(a) INTERVIEW (b) ORBI

Fig. 2. Comparison of coding efficiency

In depth map coding with H.264, there are much more skipped macroblocks and 16x16 modes than the case of the texture video. These modes form approximately 95% when the quantization parameter (QP) is 40. This is mainly due to simplicity of the depth map. On the other hand, sub-macroblock modes, such as 8x8, 4x8, 8x4 and 4x4, rarely occur in depth map coding. Coding with sub-macroblock modes increases the bit rate due to numerous motion vectors. In depth map coding, a large number of macroblocks in the inter frame are encoded in 16x16 or 4x4 intra modes. The intra mode is a prediction scheme which exploits the neighboring pixels around the block to be coded. The intra mode works well for smooth images like the depth map and saves the coding bits for motion vectors.

In the inter frame, bits are composed of seven components: header, mode, motion information, coded block pattern (CBP), residual data, delta QP, and stuffing bits. Residual data in the depth map usually takes a relatively small portion, which means that it is predicted well with motion estimation or intra prediction. In addition, bits for motion information take a large portion (more than 40%) even though most macroblocks are encoded in large sizes of blocks and intra coding.

3 Similarity Analysis for Motion Information

In general, the texture video and the depth map sequence have similar characteristics. For example, boundaries of objects coincide and directions of object movements are the same in both sequences. During the motion estimation operation in the texture video, the motion vector is estimated by

$$J_{motion}(MV, REF \mid \lambda_{motion}) = SAD(MV, REF) + MCOST(MV, REF) \qquad (1)$$

where *MV* is the motion vector and *REF* is the reference frame for motion estimation. The motion cost (*MCOST*), which indicates the number of coding bits for the motion vector, is defined by

$$MCOST = \lambda_{motion} \frac{(\text{mvbits}[4c_x - p_x] + \text{mvbits}[4c_y - p_y])}{2^{16}} \qquad (2)$$

where λ_{motion} is the Lagrangian multiplier of the motion vector and it is determined based on the quantization parameter (QP) by $\sqrt{0.85\times 2^{(QP-12)/3}}$. In Eq. (2), (c_x, c_y) is the position of the actual candidate motion vector and (p_x, p_y) is the position of the predicted motion vector obtained by the left, upper and upper-right blocks. mvbits[·] is an array which contains expected bits for encoding of the motion vector, as shown in Eq. (3). The farther (c_x, c_y) is away from (p_x, p_y), the more the estimated cost for the motion vector increases.

$$\text{mvbits}[\pm i] = 2i + 1 \quad i = 0, 1, 2, \cdots, 3 + 2 \times \lceil \log_2(\#\ \text{of positions} + 1) \rceil \qquad (3)$$

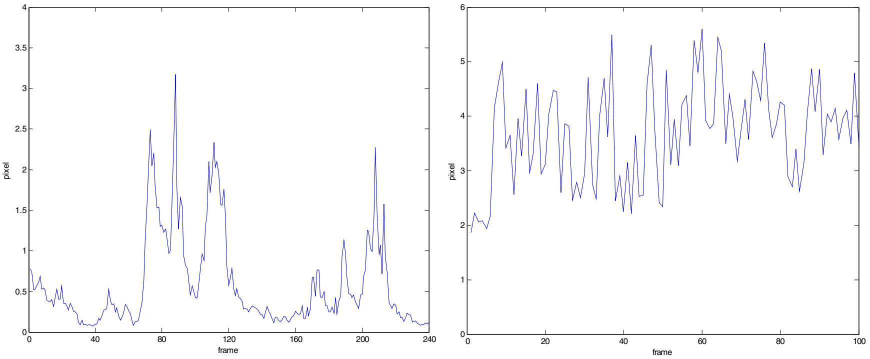
Therefore, the motion cost is carried out as a smoothness constraint. This means that motion vectors are not random and have similar values as neighboring motion vectors even when the size of the block is 4x4. Because of this property of motion vectors, the structure of objects in the depth map is maintained and some blocks may have incorrect motion compensation.

Comparing the similarity of motion vectors between the texture video and the depth map sequence, we define a difference measure by the average distance of two motion vectors in the frame:

$$\text{dist}_{\text{frame}} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left\| \mathbf{t}_{i,j} - \mathbf{d}_{i,j} \right\| \qquad (4)$$

where **t** is the motion vector for each 4×4 block of the texture video and **d** is that of the depth map sequence.

Figure 3 shows the difference of motion vectors per frame of two test sequences, INTERVIEW and ORBI. The maximum search range is 32. While the average difference of the INTERVIEW sequence with a small motion is about 0.59 pixels, the ORBI sequence whose motion is relatively large because of camera motion has about 3.77 pixels.



(a) INTERVIEW (b) ORBI

Fig. 3. Average difference of motion information per frame

4 Motion Information Sharing Algorithm

When we encode the depth map sequence, we can share the motion information of the corresponding texture video by exploiting the similarity of motion vectors between two sequences. The proposed algorithm consists of three stages, as illustrated in Fig. 4. Initially, we need to decode coding modes and associated motion vectors of the corresponding texture video to use the motion information for depth map sequence coding. From the decoded motion information, we generate various modes and associated motion vectors. Then, we select an optimal mode among the generated candidate modes based on the rate-distortion theory.

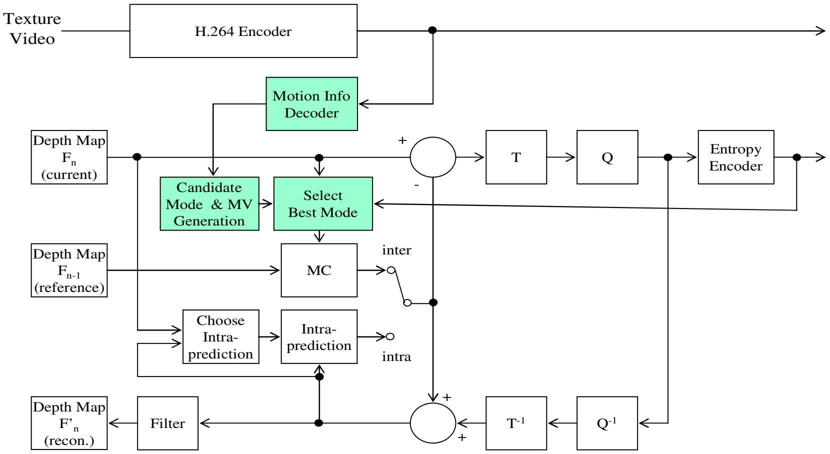


Fig. 4. Block diagram of the proposed system

4.1 Decoding of Motion Information

Inter prediction supports a range of block sizes from 16×16 down to 4×4. In addition to the inter prediction, intra prediction is also available in the inter frame and has two modes: 16×16 mode and 4×4 mode. If prediction with neighboring pixels provides better performance than prediction with motion vectors, the intra mode is selected. However, intra coding rarely occurs in the texture video. For most QPs, the frequency of the intra mode is about 1.0% or less. Since intra prediction does not have motion information, direct use of motion vectors in depth map sequence coding is impossible. Instead, the motion vector can be generated from motion vectors of the neighboring blocks. In our experiment, the median of motion vectors for left(**A**), upper(**B**), and upper right(**C**) blocks has shown a good performance.

$$(p_x, p_y) = MEDIAN(\mathbf{A}, \mathbf{B}, \mathbf{C}) \tag{5}$$

4.2 Candidate Mode Generation

Since modes and motion vectors of the texture video were optimized for the texture video, they should be adjusted to fit into the depth map sequence coding. For this purpose, we need to generate various candidate modes and motion vectors. Generation of candidate modes can be divided into two operations: merge and split operations.

Figure 5 shows an example of generating candidate modes from the decoded mode. If the mode of the texture video consists of smaller partitions, a merge operation is performed and generates larger sizes of modes. If the mode of the texture video is larger than the size of the current mode to be generated, a split operation is performed and generates several smaller sizes of modes using the neighboring blocks.

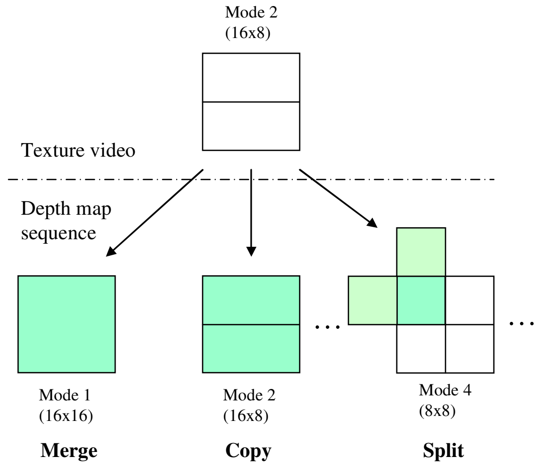


Fig. 5. Example of candidate mode generation

The motion vector for each mode is obtained by

Copy : If the mode is the same as that of the texture video, motion vectors are simply copied from the texture video.

Merge : The average of the motion vectors within the size of the current mode.

Split : The average of the motion vectors within the size of the current mode and those of left and upper blocks to avoid generating duplicate motion vectors.

4.3 Rate-Distortion Optimization

The best mode is selected from the newly generated inter modes, SKIP mode, and intra modes based on the following rate-distortion cost.

$$\mathbf{I}^* = \arg\min_{\mathbf{I}} D(\mathbf{S}, \mathbf{I} \mid \lambda) + \lambda \cdot R(\mathbf{S}, \mathbf{I} \mid \lambda) \tag{6}$$

where **S** is the set of blocks to be coded, D is the distortion, R is the bit rate, **I** and **I*** are encoding parameters and the best encoding parameters, respectively. In Eq. (6), λ is the Lagrangian multiplier of the mode. R includes the bits of the header, CBP, mode, motion information, and residual data. By skipping the bits for motion information, the best encoding parameters are selected with more weighting on distortion. Thus, some increase in PSNR values is expected. Moreover, when motion compensation is performed with wrong motion vectors, error propagation can be blocked due to the intra mode in the inter frame.

However, the proposed algorithm does not always show better performance at all bitrates. Sharing of motion vectors may be suboptimal in terms of the prediction error criterion. Therefore, residual data can be increased compared to direct coding of the depth map separately. Our approach is advantageous only when coding bits of the residual data are less than bit savings obtained by no coding of motion vectors. In other words, coding performance is good at low bitrates where the residual data is roughly quantized. This scheme is effective even when the depth map image, which is considerably degraded at low bit rates, is used in synthesizing views [7].

5 Experimental Results and Analysis

We have implemented the proposed algorithm into JM reference software 9.7 and our simulation conditions are summarized in Table 1. Two types of test sequences have been used. The depth map sequences of ORBI and INTERVIEW [8] were captured by an infrared range camera, so-called Zcam™ [9]. On the other hand, the depth map sequences of BREAKDANCERS and BALLET [10] were calculated by a state-of-art stereo matching algorithm from multiple scenes. In our experiment, we have compared the proposed scheme to the original coding scheme where the depth map sequence is coded separately. Motion vectors have been taken from various QPs (28, 32, 36, 40) in the corresponding texture video. Since the depth map sequence for 3D-TV generally shows high quality at low bitrates, we have evaluated our proposed algorithm in such high QPs (more than 40).

Figure 6 shows the encoding times of 100 frames of the texture video and the depth map sequence, respectively. Since the proposed algorithm omits the motion estimation operation that has high computational complexity, the encoding time of the proposed algorithm was about 60% of the original scheme.

Figure 7 shows R-D curves of the original scheme and the proposed scheme. At low bitrates, coding efficiency has been improved up to 1dB, which shows the similar tendency on both types of test sequences. In particular, improvement of coding efficiency is easily observed in sequences having a large motion, because sequences with a large motion save a lot of bits by not sending motion information of the depth map sequence. In addition, motion vectors obtained from the texture video that is encoded in higher quality have shown better results. However, at high bitrates, coding performance has significantly fallen due to the precise quantization of increased residual data.

Table 1. Simulation conditions

Number of Frames	100
Search Range	±32
Number of Reference Frames	1
Sequence Type	IPPP
Entropy Coding Method	CABAC
RD Optimization	High Complexity Mode
I Slice Insertion	0.5 sec

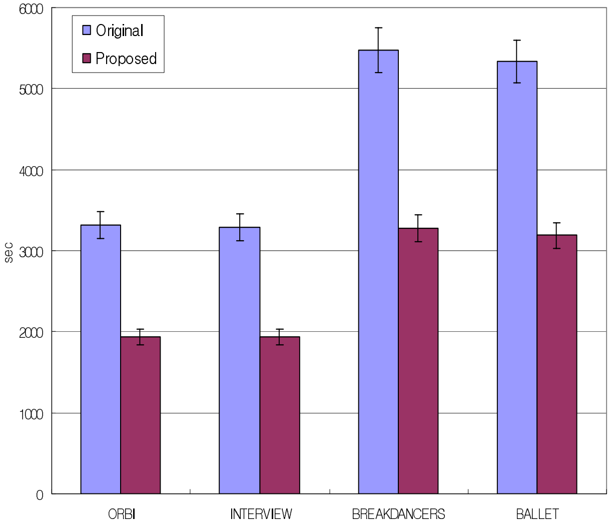
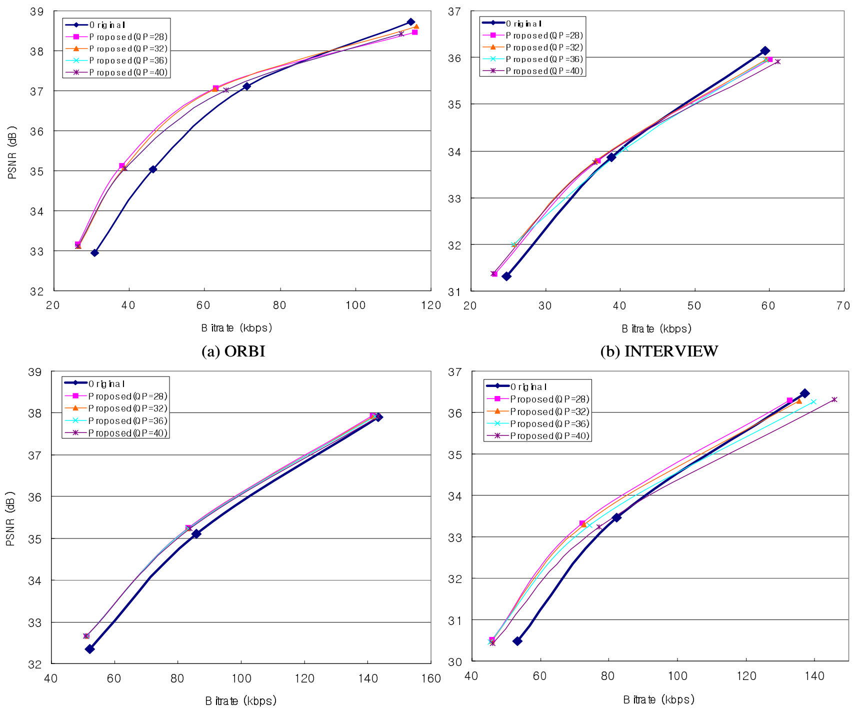


Fig. 6. Comparison of encoding times



(a) ORBI

(b) INTERVIEW

(c) BREAKDANCERS

(d) BALLET

Fig. 7. Performance comparison

6 Conclusions

In this paper, we have proposed a new H.264-based coding algorithm for the depth map sequence using motion information of the corresponding texture video. Although pixel values in both sequences are different, boundaries of objects in the scene coincide and directions of object movements are very similar. Besides, when estimating the motion vectors in the texture video, H.264 considers the cost for coding motion vectors. Hence, the structure of objects tends to be maintained. These features allow the motion vectors of two sequences to be similar. In order to share motion vectors in a proper way, we have generated various candidate modes and motion vectors from the decoded modes and motion vectors of the texture video. We then select one among those candidates based on the rate-distortion optimization. Our experimental results have demonstrated that the proposed scheme reduces the complexity up to 60% on average of the original scheme where the depth map sequence and the texture video are encoded separately. Coding efficiency has been improved up to 1dB at low bitrates. However, the proposed scheme does not always provide improved performance at high bit rates. Sometimes at high bitrates, coding performance has been rather reduced due to precise coding of increased residual data. Therefore, the proposed scheme is effective when fast encoding is required at low bit rates.

Acknowledgements

This work was supported in part by the Information Technology Research Center (ITRC) through the Realistic Broadcasting Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

References

[1] Redert, A., Op de Beeck, M., Fehn, C., IJsselsteijn, W., Pollefeys, M., Van Gool, L., Ofek, E., Sexton, I., Surman, P.: ATTEST–Advanced Three-Dimensional Television System Technologies. Proc. of International Symposium on 3D Data Processing (2002) 313–319

[2] Chai, B., Sehuraman, S., Hatrack, P.: Mesh-based Depth Map Compression and Transmission for Real-time View-based Rendering. Proc. of International Conference on Image Processing (2001)

[3] Grewatsch, S., Muller, E.: Fast Mesh-based Coding of Depth Map Sequences for Efficient 3D-Video Reproduction Using OpenGL. Visualization, Imaging, and Image Processing (2005)

[4] Fehn, C.: Depth-image-based Rendering (DIBR), Compression and Transmission for a New Approach on 3D TV. Proc. of SPIE Conf. Stereoscopic Displays and Virtual Reality Systems XI Vol. 5291 (2004) 93–104

[5] Grewatsch, S., Muller, E.: Evaluation of Motion Compensation and Coding Strategies for Compression of Depth Map Sequences. 49th SPIE's Annual Meeting (2004)

[6] Grewatsch, S., Muller, E.: Sharing of Motion Vectors in 3D Video Coding. Proc. of International Conference on Image Processing (2004)

[7] Fehn, C.: A 3D-TV Approach Using Depth-Image-Based Rendering (DIBR). Visualization, Imaging, and Image Processing (2003) 482-487

[8] Fehn, C., Schuur, K., Feldmann, I., Kauff, P., Smolic, A.: Distribution of ATTEST test sequences for EE4 in MPEG 3DAV. ISO/IEC JTC1/SC29/WG11 M9219 (2002)

[9] Iddan, G., Yahav, G.: 3D Imaging in the Studio. SPIE's Videometrics and Optical Methods for 3-D Shape Measurement Vol. 7 (2003) 48-55

[10] Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality Video View Interpolation Using a Layered Representation. ACM Transaction on Graphics Vol. 23 No. 3 (2004) 598-606

[11] Wiegand, T., Sullivan G., Bjontegaard, G., Luthra, A.: Overview of the H.264 Video Coding Standard. IEEE Transaction on Circuits and Systems for Video Technology Vol. 13 No. 7 (2003) 560-576

[12] Wedi, T.: Motion Compensation in H.264. IEEE Transaction on Circuits and Systems for Video Technology Vol. 13 No. 7 (2003) 577-586

[13] Fehn, C., Hopf, K., Quante, Q.: Key Technologies for an Advanced 3D-TV System. Proceedings of SPIE Three-Dimensional TV, Video and Display, (2004) 66-80

A Novel Macroblock-Layer Rate-Distortion Model for Rate Control Scheme of H.264/AVC

Tsung-Han Chiang and Sheng-Jyh Wang

Department of Electronics Engineering & Institute of Electronics,
National Chiao Tung University, Hsin Chu, Taiwan, R.O.C.
jkc.ee93g@nctu.edu.tw, shengjyh@cc.nctu.edu.tw

Abstract. The purpose of rate control is to adjust an encoder so that the amount of encoded bits can match the amount of desired bits. In this paper, we focus on the rate control of an H.264/AVC encoder. We first analyze the relation between the quantization parameter, MAD, and the coded bit number. We also analyze the encoding of header bits. Based on these analyses, we build up a new rate-distortion model. Moreover, we use motion vectors to predict the MAD value. Based on these modifications, we may better predict the quantization parameter and the amount of encoded bits. In experiments, our approach not only improves the stability of encoder buffer but also makes an obvious improvement of visual quality.

Keywords: Rate control, rate-distortion model, H.264/AVC.

1 Introduction

Rate control plays an important role in video encoders. The purpose of rate control is to control the amount of the encoded bits. Without rate control, the client buffer may face underflow or overflow due to the mismatch between the source bit rate and the available channel bandwidth for delivering a compressed bitstream. Existing video coding standards usually have their own non-normative rate control schemes during the standardization process. For example, there exists a rate control scheme in the JM (Joint Model) of H.264/AVC.

Fig. 1 shows a block diagram of video encoding. Generally speaking, we adjust the QP value to control the compressed data size. The factors that may affect the determination of the QP values include channel bandwidth, buffer fullness, and video complexity. In terms of the unit of the rate control operation, rate control schemes can be classified into macroblock-, slice-, or frame-layer rate control. There exist several rate control schemes for video coding standards, such as the rate control schemes in the TM5 [1] of MPEG-2, in the TMN8 [2] of H.263, and in the VM-18 [3] of MPEG-4. These rate control schemes usually resolve two main problems: the bit allocation problem to predict the coded bits, and the parameter adjustment problem to properly encode each unit based on the amount of allocated bits.

The bit allocation in rate control is usually associated with a buffer model. The quantization parameter adjustment is used to find the relation between the bit rate and

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 908 – 918, 2006.

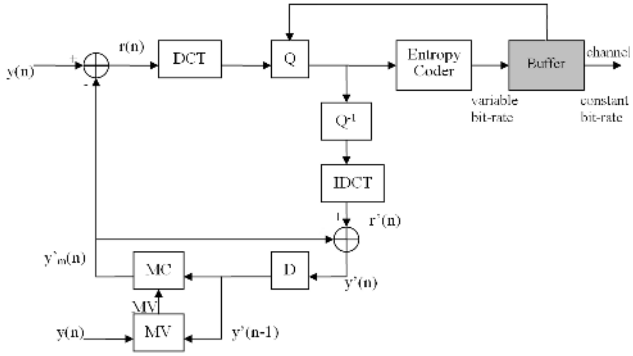


Fig. 1. Block diagram of an encoder with rate control

the quantization parameter. This relation is usually defined by a rate-distortion (R-D) model. A commonly used model in H.264/AVC is the quadratic rate-distortion model proposed by Chiang and Zhang [4] which uses the image complexity and the target bits to decide the QP value. Besides [4], many other methods have been proposed to predict the QP value for H.264/AVC, hoping to provide better coding performance and better video quality [5].

In this paper, we propose a more accurate R-D model for the adjustment of quantization parameter. In bit allocation, we perform macroblock-level bit allocation. The remainder of this paper is organized as follows. In Section 2, the backgrounds of H.264/AVC standard and rate control are briefly reviewed. In Section 3, our R-D model is statistically and theoretically analyzed. Our bit allocation is then discussed in Section 4. Finally, Section 5 concludes this paper.

2 Modified Rate-Distortion Model for H.264/AVC

In the Joint Model of H.264/AVC, the rate-distortion model adopts the quadratic rate-distortion model, which is proposed by Chiang and Zhang [4] for MPEG4. This model is described as

$$T = c_1 \times \frac{\tilde{\sigma}}{Q_{step}} + c_2 \times \frac{\tilde{\sigma}}{Q_{step}^2} + m \,. \tag{1}$$

where T is the target bits which denote the predicted number of coded bits, m is the predicted number of the header bits, σ means the MAD value, and c_1 and c_2 are some constant coefficients. Using this model, we can estimate the quantization stepsize. After a simple conversion, the QP value can be computed. According to Equation (1), the coded bits can be divided into two parts. The first two items represent the texture bits, while the last item represents the header bits. In this paper, we formulate a new R-D model according to the relation between texture bits and the quantization parameter. Since the header bits are also important in the R-D model, we also investigate the relation between the header bits and the macroblock mode. Finally, we build a modified R-D model to provide an improved performance in rate control.

2.1 Proposed Rate-Distortion Model

First, we are interested in modeling the relation between the number of coded texture bits and the quantization parameter, assuming the residual (MAD) of the macroblock is fixed. Fig. 2 shows the plot of the number of generated bits with respect to QP. It can be seen in Fig. 2 that the relation between the number of generated bits and the QP value can be partially approximated by a second-order polynomial. That is,

$$Bits = a \times QP^2 + b \times QP + c\ . \tag{2}$$

Fig. 3 describes the curve fitting results based on the second order polynomial. It can be seen that the second order polynomial fits the relationship pretty well. Then, we try to find the relation between MAD and the coefficients in Equation (2). Fig. 4 plots the relation between the coefficient "a" and MAD. Based on this experiment, we model the relation to be linear. The linear model is expressed as

$$a = \max\left(0, x_1 + x_2 \times MAD\right). \tag{3}$$

The lower bound is to avoid getting a negative "a".

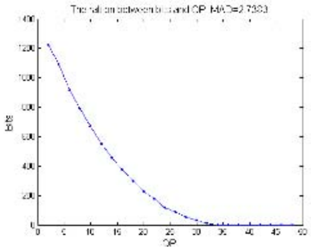


Fig. 2. Relation between coded texture bits and QP

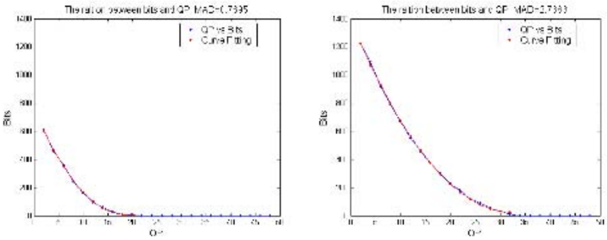


Fig. 3. The relation between the number of coded texture bits and QP, when MAD=0.7695 and MAD=2.7383. The pink curves show the fitting results based on a second-order polynomial.

Similarly, we can find the relationship between the coefficient "b" and MAD. Fig. 5 shows that b is approximately equal to the constant -90 for various choices of MAD. Even

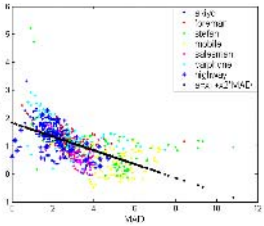


Fig. 4. The relation between the coefficient "a" and MAD in different sequences

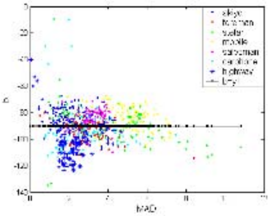


Fig. 5. The relation between the coefficient "b" and MAD in different sequences

though the variance of b is not very small, we still treat "b" as a constant in our model to simplify the problem. Regarding the coefficient "c", we can see in Fig. 6 that there is an apparent relationship between "c" and MAD. We have tried several curve models to fit this relationship and finally reached the simple model expressed in equation (4). The fitting result is also shown in Fig. 6.

$$c = z_1 \times MAD + z_2 \times \sqrt{MAD} \ . \tag{4}$$

Based on Equations (3) and (4), we can rewrite Equation (2) as

$$Bits = k_1 \times MAD \times QP^2 + k_2 \times MAD + k_3 \times QP^2 + k_4 \times \sqrt{MAD} + k_5 \times QP \ . \tag{5}$$

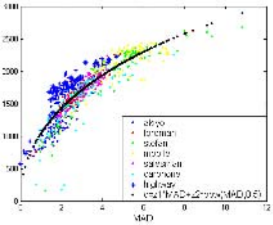


Fig. 6. The relation between the coefficient "c" and MAD in different sequences

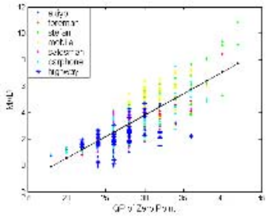


Fig. 7. The relation between MAD and the "zero point"

The equation (5) needs one extra constraint. This is because this model is used only for non-zero coded bits. Observing Fig. 2, we can find that when the QP value is large enough, the number of coded bits become 0. For example, the number of coded bits becomes 0 when QP is larger than 34 in Fig. 2. We name this turning point as the "zero point". In Fig. 7, we show the relationship between MAD and the "zero point". Based on Fig. 7, we describe the relation between MAD and the "zero point" as a linear model and express the relationship as

$$MAD = h_1 + h_2 \times QP \ . \tag{6}$$

Then, we add on the constraint that MAD has to be larger than some value. The equation (5) is then rewritten as

$$Bits = k_1 \times MAD \times QP'^2 + k_2 \times MAD + k_3 \times QP'^2 + k_4 \times \sqrt{MAD} + k_5 \times QP' \cdot$$

$$QP = \begin{cases} QP' & QP' \le \dfrac{MAD - h_1}{h_2} \times \sigma \\ QP' - 1 & QP' > \dfrac{MAD - h_1}{h_2} \times \sigma \end{cases} \tag{7}$$

where σ is a parameter that must be more than one.

In this new model, we'll have to calculate the model coefficients k1, k2, k3, k4 and k5. Here, we use a linear regression technique to find these coefficients. Based on this model, we then formulate our R-D model for H.264/AVC at the macroblock level.

2.2 The Relation Between Header Bits and Macroblock Mode

In H.264/AVC, inter-prediction coding has several macroblock modes. The number of macroblock header bits is varying in different macroblock modes. In Fig. 8, we show that the header bits of 16×16, 16×8 and 8×16 macroblock modes are basically irrelative to MAD, but the header bits of the 8×8 mode are proportional to the MAD value of the macroblock. Moreover, the variances of the header bits in 16×16, 16×8 and 8×16 modes are very small, but it is not the case in the 8×8 mode.

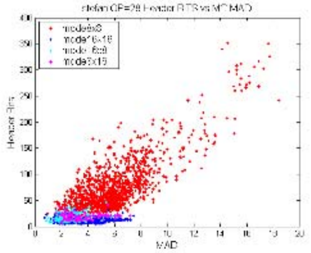


Fig. 8. The relation between header bits and the MAD value of macroblocks

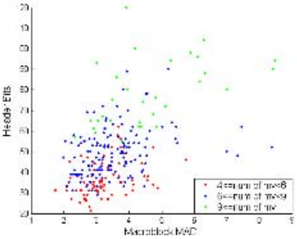


Fig. 9. The relation between macroblock header bits, number of motion vector, and macroblock MAD

In the 8×8 mode, the motion vector data are variable. This is because an 8×8 block can be divided into 8×4, 4×8, or 4×4 modes. A smaller block size causes more motion vectors and thus more header bits. In Fig. 9, we can observe such a relation. Hence, a larger MAD has a larger chance to encode more motion vectors. Based on this observation, we model the relation between the 8×8 header bits and the MAD value of the macroblock in terms of a linear model:

$$Hbits_{8\times8} = a \times MAD + b \ . \tag{8}$$

where a and b are coefficients of this model.

For other modes, such as 16×16, 16×8 and 8×16 modes, we simply estimate the header bits of the current macroblock to be the averaged value of the previous macroblock. Using this header bits prediction method, we may reformulate Equation as

$$Bits = k_1 \times MAD \times QP'^2 + k_2 \times MAD + k_3 \times QP'^2 + k_4 \times \sqrt{MAD} + k_5 \times QP' + Hbits$$

$$QP = \begin{cases} QP' & QP' \le \dfrac{MAD - h_1}{h_2} \times \sigma \\ QP' - 1 & QP' > \dfrac{MAD - h_1}{h_2} \times \sigma \end{cases} \tag{9}$$

where the coefficients are computed based on the linear regression technique over the already encoded data.

3 Bit Allocation at Macroblock Level

In the rate control scheme of the JM model in H.264/AVC, if the number of basic unit in a picture is one, we need to determine the bit allocation for each basic unit. If the basic unit is a macroblock, we perform bit allocation for each macroblock. In JM [8], the original method is defined as

$$T_{mb}(j,k) = \frac{pMAD(j,k)^2}{\sum_k pMAD(j,k)^2} \times T(j) \cdot \tag{10}$$

where j denotes the frame index, T(j) represents the target bits of the jth frame, k denotes the index of the encoded macroblock, and $T_{mb}(j,k)$ represents the target bits of the kth macroblock at the jth frame. The symbol pMAD denotes the predicted MAD of a macroblock. This equation means that a macroblock with a larger MAD value needs more bits to be coded. Here, the prediction of MAD is based on a linear model:

$$pMAD(j,k) = a_1 \times MAD(j-1,k) + a_2 \cdot \tag{11}$$

where pMAD means the predicted MAD value, MAD means the computed MAD value, and a_1 and a_2 are coefficients.

However, when a macroblock is under movement, the corresponding residual data should be in motion too. Hence, the MAD of that macroblock no longer stays at the same place. Fig. 10 illustrates such a situation where the movement of macroblocks causes the MAD patterns in the current frame to be shifted to some other places in the next frame.

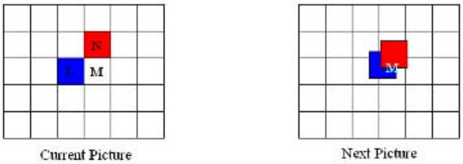


Fig. 10. The illustration of macroblock motion

To modify the MAD prediction at the macroblock level, we check the motion vector of each macroblock and move the MAD of each macroblock to the new location accordingly. After moving all the macroblocks, we calculate the averaged MAD to represent the predicted MAD value. Like the case in Fig. 10, the MAD of macroblock M in the next picture can be computed as:

$$pMAD_{t+1}(M) = \frac{w_N \times MAD_t(N) + w_L \times MAD_t(L) + w_M \times MAD_t(M)}{w_N + w_L + w_M} . \tag{12}$$

where pMAD means the predicted MAD, w_i represents the weighting coefficient of a macroblock. This modification not only improves video quality but also reduces the probability of buffer fullness.

4 Proposed Rate Control Scheme for H.264/AVC

The proposed rate control scheme for H.264/AVC contains three parts. The first part is the GOP-level rate control, where the initial QP and the definition of buffer fullness of this GOP are determined. The second part is the picture-level rate control, where the QP of each frame is determined. Finally, the basic-unit-level rate control determines the QP value of each basic unit. In Table 1, we list the symbols used in this section.

In the GOP-level rate control, the expected available bits for GOP encoding are computed. The initial QP value is determined according to the average QP value of the previous GOP. If this is the first GOP, the initial QP value is determined according to the channel bandwidth and video resolution.

The picture-level rate control is further divided into two stages, pre-encoding stage and post-encoding stage. In the pre-encoding stage, we need to determine the target bits of this picture. The target bits are predicted in two aspects. One is according to buffer fullness and channel bit rate. The other is determined by the remaining bits of this GOP. That is, we have

$$\tilde{T}_i(j) = \frac{R_i(j)}{f} + \gamma \times \left(S_i(j) - V_i(j) \right) . \tag{13}$$

$$\hat{T}_i(j) = \frac{B_i(j)}{N_{p,r}} . \tag{14}$$

The real target bits are computed by combining $\tilde{T}_i(j)$ and $\hat{T}_i(j)$ together. In this paper, we choose

$$T_i(j) = 0.5 \times \hat{T}_i(j) + 0.5 \times \tilde{T}_i(j) . \tag{15}$$

On the other hand, the post-encoding stage adds the actual encoded bits into the buffer and makes sure whether the buffer overflows.

Finally, for the basic-unit-level rate control, we first use Equation (12) to predict the MAD value of each macroblock and compute the target bits of each basic unit based on Equation (10). Then, we determine the QP value of each basic unit according to the

Table 1. Summary of Symbols

Parameter Name	Definition
i	The index of GOP
j	The index of Picture in each GOP
f	Frame rate
$R_i(j)$	Channel bit rate
$N_p(i\text{-}1)$	Total number of stored pictures in the (i-1)th GOP
$B_i(j)$	The bits for the rest pictures in this GOP
$S_i(j)$	Target buffer level
$V_i(j)$	Buffer fullness
$\tilde{T}_i(j)$	The delta target bits
$\hat{T}_i(j)$	The hat target bits
$T_i(j)$	Target bits
$N_{p,r}$	The number of the remaining stored pictures

R-D model expressed in Equation (9). After determining the QP value, we execute the RDO encoding in each macroblock. The last step is to update the coefficients by the linear regression method.

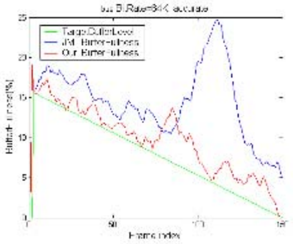
5 Experimental Results

In order to evaluate the performance of the proposed algorithm, we performed several experiments on a few test sequences. The definitions of coding parameters are listed as followings.

Profile: Baseline
Frame Rate: 30 frame per sec
Buffer size: 0.5×bit rate
Frame Size: QCIF
GOP: IPPP…
Basic Unit: one macroblock
RDO: On
Bit Rate: 64K, 128K

Table 2. The precision of prediction model

Sequence	Bit Rate	Texture-bit			Header-bit		
		Error (MAD)		Improvement (JM-Our)/JM	Error (MAD)		Improvement (JM-Our)/JM
		JM	Our		JM	Our	
Bus	64K	569.7	193.7	**0.66**	230.50	195.20	**0.15**
	128K	224.2	150.5	**0.33**	448.91	307.23	**0.32**
Flower	64K	1185.9	995.2	**0.16**	346.00	220.79	**0.36**
	128K	1965.9	700.7	**0.64**	466.33	303.87	**0.34**



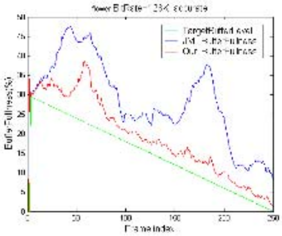
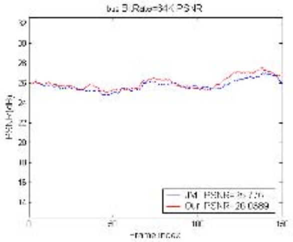


Fig. 11. The buffer fullness curves when coding the “bus” sequence with the bit rate = 64K

Fig. 12. The buffer fullness curves when coding the “flower” sequence with the bit rate = 128K



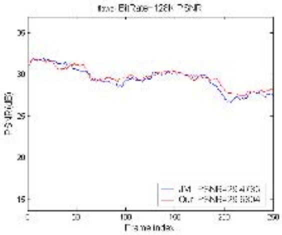


Fig. 13. The PSNR curve when coding the “bus” sequence with the bit rate = 64K

Fig. 14. The PSNR curve when coding the “flower” sequence with the bit rate = 128K

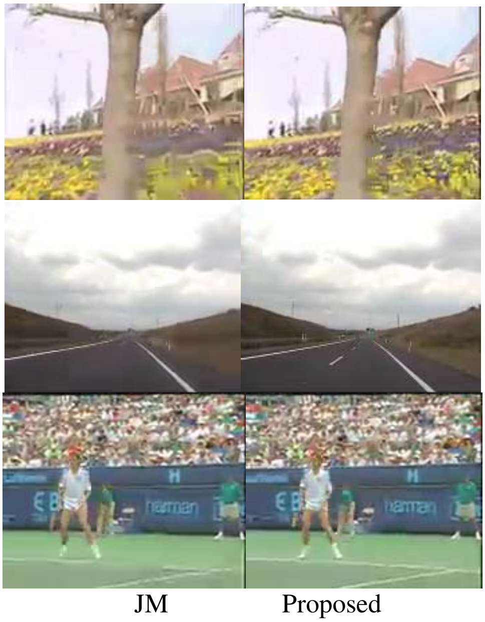


Fig. 15. Comparison of visual quality for the 220-th frame of the “flower” sequence result when the bit rate is 64K

We first compare the precision of the proposed R-D model. We divided the coded bit into two parts, texture bits and header bits. The comparisons with respect to the JM of H.264/AVC are listed in Table 2. The "error" is defined as the mean of absolute difference (MAD) between the actual coded bits and the targets bits (predicted bit number).

In Table 2, the improvement is apparent. As shown in Fig. 11 and Fig. 12, our results are closer to the target buffer level. This improvement also causes the increase of PSNR values. Beside the proposed R-D model, the modified MAD prediction also causes the increase of PSNR values in the first few frames of each GOP. The average PSNR gain is 0.26dB and 0.2dB, respectively. Besides the PSNR gain, our proposed method can provide improved visual quality. In Fig. 15, we can see obvious differences, like the flowers in the garden. The improvement is due to the fact that we have reduced the over-use of coded bits during the coding of the upper portion of the pictures. Hence, adequate bits are left for the coding of the bottom portion of the pictures.

6 Conclusion

In this paper, we propose a new rate-distortion model for the baseline profile of H.264/AVC. Using the modified R-D model, we can improve the accuracy of rate control and the performance of visual quality, especially at low bitrates. The relation between the header bit and the MAD value has also been discussed. In the basic unit level, we use an instinctive method to predict the MAD value. Utilizing the bit allocation at the macroblock-level, we can achieve better performance over bit allocation and visual quality.

Acknowledgement

This research was supported by National Science Council of the Republic of China under Grant Number NSC-94-2219-E-009-017.

References

[1] ISO/IEC JTCI/SC29/WG11, *Test Model* 5, 1993.

[2] ITU-T/SG15, *Video Codec Test Model*, TMN8, Portland, June 1997.

[3] MPEG-4 Video Verification Model V18.0, Coding of Moving Pictures and Audio N3908, ISO/IEC, JTC1/SC29/WG11, Jan. 2001.

[4] H. J. Lee and T. H. Chiang and Y.-Q. Zhang, "Scalable rate control for MPEG-4 video," *IEEE Trans. on Circuits. Syst. Video Technol.*, vol.10, No.6, pp. 878-894, Sept. 2000.

[5] Siwei Ma, Wen Gao, Feng Wu and Yan Lu, "Rate Control for JVT Video Coding Scheme with HRD Considerations," *IEEE International Conference on Image Processing*. vol. 3, pp.793-796, Sept. 2003.

[6] Thung-Hiung Tsai and Jin-Jang Leou, "A Rate Control scheme for H.264 Video Transmission," *IEEE International Conference on Multimedia and Expo (ICME)*, vol.2, pp.1359-1368, June 2004.

[7] Satoshi Miyaji, Yasuhiro Takishima and Yoshinori Hatori, “A Novel Rate Control Method for H.264 Video Coding.” *IEEE International Conference on Image Processing*. vol. 2, pp.II-309-12, 11-14 Sept. 2005.

[8] Siwei Ma and Wen Gao, “Rate-Distortion Analysis for H.264/AVC Video Coding and its Application to Rate Control.” *IEEE Trans. on Circuits and Syst. Video Technol.*, vol.15, No.12, pp.1533-1544, Dec. 2005.

[9] Z. Li et al., "Adaptive Basic Unit Layer Rate Control for JVT," *JVT-G012, 7th Meeting*: Pattaya, Thailand, March 2003.

A Geometry-Driven Hierarchical Compression Technique for Triangle Meshes

Chang-Min Chou[1,2] and Din-Chang Tseng[1]

[1] Institute of Computer Science and Information Engineering, National Central University, No. 300, Jungda Rd., Jhongli City, Taoyuan, Taiwan 320
[2] Department of Electronic Engineering, Ching Yun University, No. 229 Cheng-Shing Rd., Jhongli City, Taoyuan, Taiwan 320
changmin@cyu.edu.tw
tsengdc@ip.csie.ncu.edu.tw

Abstract. A geometry-driven hierarchical compression technique for triangle meshes is proposed such that the compressed 3D models can be efficiently transmitted in a multi-resolution manner. In 3D progressive compression, we usually simplify the finest 3D model to the coarsest mesh vertex by vertex and thus the original model can be reconstructed from the coarsest mesh by operating vertex-split operations in the inversed vertex simplification order. In general, the cost for the vertex-split operations will be increased as the mesh grows. In this paper, we propose a hierarchical compression scheme to keep the cost of the vertex-split operations being independent to the size of the mesh. In addition, we propose a geometry-driven technique, which predicts the connectivity relationship of vertices based on their geometry coordinates, to compress the connectivity information efficiently. The experimental results show the efficiency of our scheme.

Keywords: Mesh compression, progressive compression, progressive mesh.

1 Introduction

3D models have been represented by triangle meshes in computer graphics for a long time. In recent years, the issue of 3D model compression, multi-resolution, and progressive transmission gain more attention due to the usage of advanced scanning devices and the growth of Internet. Transmission of 3D models over Internet is relatively slow for complex models with many fine details. A progressive representation is thus appreciated due to its ability to enhance high performance interactivity with large model transmission. Furthermore, in progressive encoding schemes, models can be encoded as an embedded bit stream so that the receiver can terminate the transmission at any point of time to get an approximation of the model with exact bit control [9].

Typically, meshes are represented by two kinds of data: connectivity and geometry. The former represents how vertices in a mesh are connected while the latter represents the coordinates (and normals, textures, etc.) of each vertex. When considering the problem of 3D compression, for the purpose of reconstruction, the

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 919 – 928, 2006.

connectivity information has to be encoded (or compressed) losslessly while the geometry information can be encoded in a lossy manner according to the requirement of accuracy [2], [5]. As a result, it is not easy to get a high compression ratio for the connectivity information encoding, especially when the encoding task has to be done in a progressive manner.

Previous works on 3D compression can be distinguished to two major categories: single-resolution [12], [15] and multi-resolution [1], [4], [7], [9], [10], [13] compressions. Heckbert and Garland [6] gave a detailed introduction to most famous polygonal surface simplification algorithms. Peng *et al.* [11] gave a survey on most 3D mesh compression technologies up to date. Rossignac's Edgebreaker [12] and *TG* [15] are both great works for single-resolution connectivity compression of triangle meshes. Both works can achieve 1.5 ~ 2 bits/*v* or even better for connectivity compression. Valette and Prost [16] proposed a lossy to lossless progressive compression scheme for triangular meshes based on a wavelet multi-resolution theory for irregular 3D meshes. Multi-resolution techniques usually transmit a crude model followed by a series of refinement information such that the 3D model can be reconstructed progressively.

Hoppe's *PM* [7], [8] model introduced a continuous progressive compression scheme, but is space consuming in real practice. In the *PM* model, the simplification process can be described as a sequence of *edge-collapse* operations, and the reconstruction process is a sequence of the inversed *edge-collapse* operations, which are called *vertex-split* operations. These two operations are illustrated in Fig. 1. In the *PM* model, the cost for encoding each individual vertex split operation includes three major components: 1. Information that indicates the split vertex, *v*, in the previous mesh. Theoretically, it takes $\lceil \lg n \rceil$ bits to indicate the split vertex for a mesh with *n* vertices. 2. Information that specifies how a vertex *v* is split. This can be achieved by recording the two co-vertices among all incident vertices of vertex *v*. It costs C_2^k bits to record these two co-vertices, where *k* is the number of *v*'s incident vertices. 3. Information that records the geometry (coordinate) information of the new added vertices. This is usually achieved by encoding the prediction residues, which are the differences between the predicted coordinates and the actual coordinates.

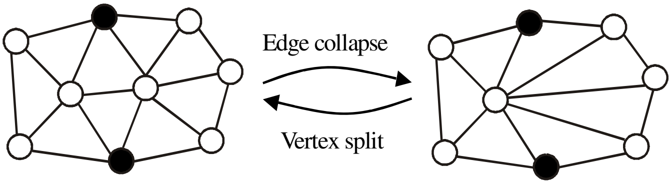


Fig. 1. 'edge collapse' and 'vertex split' operations in the *PM* model. The two black vertices are co-vertices.

Continuous progressive compression schemes usually impose a significant overhead to transmit the full resolution model when compared to the best single resolution schemes. Pajarola and Rossignac suggested a *CPM* (Compressed Progressive Meshes) [10] approach that refines the topology of the mesh in batches to eliminate the overhead caused by progressive refinement techniques and achieved great success. In this paper, we propose a hierarchical compression scheme to keep the cost of the vertex-split operations being independent to the size of the mesh. Our work is similar to the work of Pajarola and Rossignac in the hierarchical manner. However, we propose a different geometry prediction scheme and a geometry-driven connectivity prediction scheme to compress both the geometry and connectivity information efficiently.

2 Hierarchical Simplification

Most 3D geometric progressive compression algorithms first transmit a crude model, M_0, followed by a sequence of refinement information. Each refinement file contains the information of a set of vertices for reconstructing the next finer model. Fig. 2 illustrates the reconstruction procedure. Our hierarchical transmission scheme can be organized as follows:

1. A crude model, M_0, is transmitted using a single resolution compression mechanism. In our work, the *TG* [15] method is applied on this stage. We suggest M_0 to contain about 10 percents of the vertex number of the original model for the best performance in compression.
2. A sequence of refinement archive information of each hierarchy, R_1, R_2, …, R_n, are sent in order for mesh reconstruction. The refinement archives should contain both the connectivity and geometry information in each hierarchical.

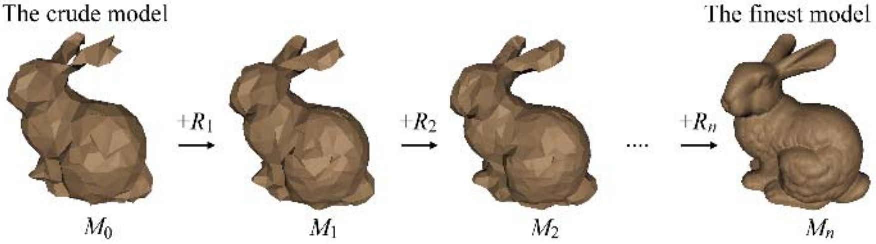


Fig. 2. The refinement reconstruction procedure

We use a ‘half-edge’ algorithm for the *edge-collapse* operation. That is, for the two vertices at the ends of the collapsed edge, we simply merge one vertex to the other, as shown in Fig. 3. For the following discussion of this paper, we’ll call the existed vertex as the parent vertex, v_p, and the disappeared vertex as the child vertex, v_c. One major goal of our hierarchical simplification mechanism is to reduce the heavy burden of indicating the split vertex. The cost for indicating the split vertex is $\lceil \lg n \rceil$ in the *PM* scheme and thus will be increased as the mesh grows. We propose a hierarchical simplification scheme that removes a set of vertices from M_i to form a simplified mesh, $M_{i\text{-}1}$. The set of removal vertices in each hierarchy must be an independent set. An independent set is a set of vertices whose removal will not affect the neighborhood relationship of any other vertex in the same set. In other words, by removing an independent set of vertices from a mesh, the final resulted mesh is independent to the order of vertex removals. To satisfy this requirement, we set three constraints in the vertex removal process of each hierarchy:

1. Boundary vertex can not be a child vertex.
2. The adjacent vertices of a child vertex cannot serve as child vertices of other vertices in the same hierarchy. Fig. 3 illustrates this constraint.
3. A vertex can serve as a parent vertex for at most one time in each hierarchy.

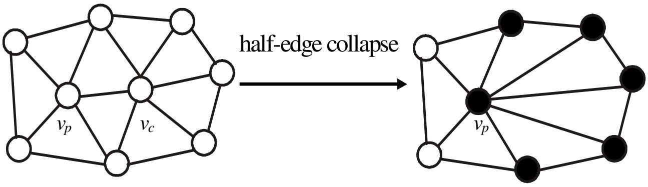


Fig. 3. Illustration for the 'half-edge' collapse operation and independent set selection constraints. Vertex v_c is merged to v_p. Once v_c is merged to v_p, the adjacent vertices of v_c (black vertices) can not serve as child vertices of any other vertex in the same hierarchy.

The proposed scheme can successfully reduce the cost for indicating the split vertices. The simplification process is described as follows:

1. Set $i = n$, where n is the number of the predefined hierarchical levels.
2. Select an independent set of vertices, V_i, from M_i.
3. Remove all vertices of V_i from $M_{i,}$ and re-triangulate the holes left by those vertex removals to form a simpler mesh $M_{i\text{-}1}$.
4. Record the refinement information R_i; set $i = i$-1.
5. Stop if $i = 0$; else go to *Step* 2.

3 Refinement Archive Encoding and Decoding

For a simplification hierarchy from M_i to $M_{i\text{-}1}$, we need to encode an archive of refinement information for the reconstruction in the receiver side. The refinement archive includes three major components: which vertices were removed (the split vertex information), the coordinates of these vertices (the geometry information), and how these vertices were connected to their neighboring vertices (the connectivity information). These three components are encoded separately and then packaged together for transmitting to the receiver side.

3.1 The Split Vertices Encoding

For each hierarchy mesh M_i, we select an independent set of vertices for simplification using the selection algorithm described in Section 2, and then execute a series of *edge-collapse* operations to produce the next hierarchy mesh $M_{i\text{-}1}$. On the other hand, a series of *vertex-split* operations should be executed on $M_{i\text{-}1}$ to reconstruct M_i in the decoding stage. We use the offsets between split vertices in the coarser mesh to record the indices of these split vertices. For example, suppose the indices of the split vertices in $M_{i\text{-}1}$ are "2, 5, 7, 11, 12…", the recorded information will be "2, 3, 2, 4, 1…", which is the first index in $M_{i\text{-}1}$ followed by a series of offsets. The split vertices are usually evenly distributed among the models, thus the offsets are highly concentrated to several integers. This characteristic implies an entropy coding method is suitable in this part. We choose an arithmetic coding method [3] for the split vertices encoding.

3.2 The Geometry Encoding

Most 3D compression schemes predict the geometry coordinates based on the connectivity relationship of vertices and then encode the prediction residues [1], [4], [10], [15]. We propose a geometry-driven technique which predicts the connectivity relationship of vertices based on their geometry coordinates. The advantage of geometry-driven compression is that the connectivity information can be compressed much more efficient. However, we still propose a prediction method based on the connectivity relationship of the previous hierarchy mesh $M_{i\text{-}1}$ to improve the geometry compression rate. The concept of our prediction method is based on the observation that the average coordinates of a parent vertex's neighboring vertices in $M_{i\text{-}1}$ is usually close to the average coordinates of it and its corresponding child vertices. That is

$$\frac{1}{\left|N(v_p)\right|}\sum_{v_j\in N(v_p)} v_j \cong \frac{1}{2}(v_p+v_c) \tag{1}$$

where $N(v_p)$ is the set of v_p's neighboring vertices and $\left|N(v_p)\right|$ is the size of $N(v_p)$. Thus, given the coordinates of v_p and its neighboring vertices in $M_{i\text{-}1}$, we can predict the coordinates of v_c as

$$v_c^{'} = \frac{2}{\left|N(v_p)\right|}\sum_{v_j\in N(v_p)} v_j - v_p \tag{2}$$

The prediction residues $(v_c - v_c^{'})$ are then encoded for geometry reconstruction. Fig. 4 illustrates the concept of the geometry prediction method. Since the geometry prediction residues (*GPR*s) tend to be very small when comparing to the actual vertex coordinates, they often distribute within a small range of the coordinate scope. According to this characteristic of *GPR*s, we encode the *GPR*s by the arithmetic coding method.

3.3 The Connectivity Encoding

For each *edge-collapse* pair v_p, v_c and their neighboring vertices, we encode the connectivity relationship based on their geometry information by the following steps:

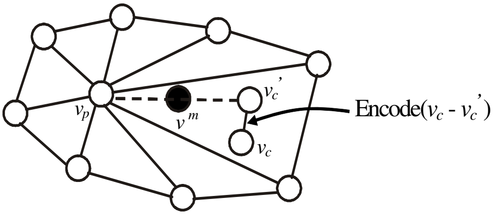


Fig. 4. The concept of the geometry prediction method. v^m is the barycenter of v_p's neighboring vertices. v_c is the actual coordinates and v_c' is the predicted coordinates of the child vertex.

1. Connect each neighboring vertex to either v_p or v_c, depending on which one is nearer (evaluated by Euclidean distance) to the neighboring vertex. This operation will leave two quadrilateral holes as shown in Fig. 5. We define the quadrilateral hole in which the $\overrightarrow{v_p v_c}$ direction is clockwise as the right quadrilateral hole, and the other one as the left quadrilateral hole. The four corner vertices of the two quadrilateral holes are named as v_p^r, v_c^r, v_p^l, and v_c^l, according to their positions, as shown in Fig. 5.

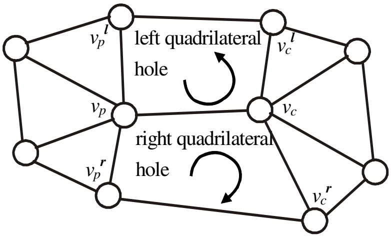


Fig. 5. The two quadrilateral holes and four corner vertices

2. Compare the produced structure with the same part of M_i by ignoring the quadrilateral holes. If they have the same structure, the *edge-collapse* pair is classified as one of the following four categories by referring the structure of M_i to compare the lengths of $\overline{v_p v_c^r}$ and $\overline{v_c v_p^r}$ in the right quadrilateral hole, and compare the lengths of $\overline{v_p v_c^l}$ and $\overline{v_c v_p^l}$ in the left quadrilateral hole:

   (*i*) *SS*: if the actual connections in M_i are shorter edges in both quadrilateral holes, as shown in Fig. 6 (a).

   (*ii*) *SL*: if the actual connections in M_i are shorter edge in the right quadrilateral hole and longer edge in the left quadrilateral hole, as shown in Fig. 6 (b).

   (*iii*) *LS*: if the actual connections in M_i are longer edge in the right quadrilateral hole and shorter edge in the left quadrilateral hole, as shown in Fig. 6 (c).

   (*iv*) *LL*: if the actual connections in M_i are longer edges in both quadrilateral holes, as shown in Fig. 6 (d).

   Otherwise (*i.e.*, the structures are different), the *edge-collapse* pair is classified to the "*Others*" category.

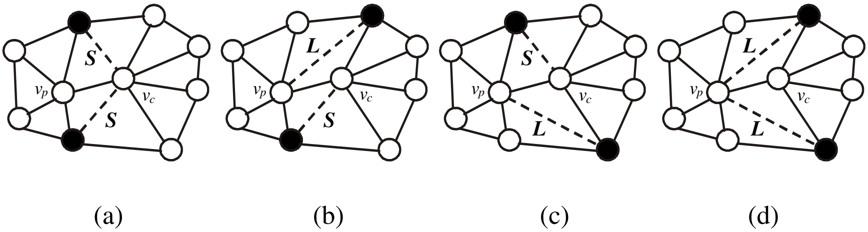


Fig. 6. The connection of (a) *SS*, (b) *SL*, (c) *LS*, and (d) *LL* categories, where "*S*" represents "shorter" and "*L*" represents "longer"

3. For vertex pairs classified to the first four categories, we directly encode their connectivity relationship as *SS*, *SL*, *LS*, and *LL*, respectively. For vertex pairs classified to the "*Others*" category, we encode the offsets between the actual connection and the *SS* condition of the two co-vertices. Fig. 7 illustrates the encoding concepts.

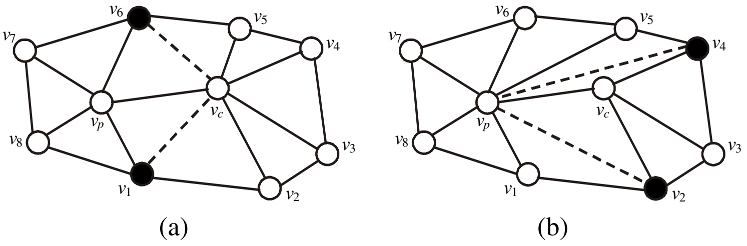


Fig. 7. The encoding of vertices belongs to the "*Others*" category. (a) The *SS* connection condition: v_1 and v_6 are the two co-vertices. (b) The actual connection condition: v_2 and v_4 are the two co-vertices. Two integers, +1 (offset between v_1 and v_2) and -2 (offset between v_6 and v_4), are encoded for reconstruction.

Table 1 shows the probability distribution of the five connectivity categories of the *Horse* model in each refinement hierarchy. Due to the uneven distribution of these categories, we encode the connectivity information by an arithmetic coding method.

Table 1. The probability distribution of five connectivity categories of the *Horse* model in each refinement hierarchy

Refinement archive	vertices	*SS*	*SL*	*LS*	*LL*	*Others*
R_1	1159	0.49	0.11	0.11	0.03	0.26
R_2	1573	0.52	0.10	0.11	0.02	0.25
R_3	2096	0.53	0.11	0.11	0.03	0.22
R_4	2821	0.57	0.13	0.11	0.03	0.16
R_5	3824	0.61	0.13	0.13	0.02	0.11
R_6	5093	0.73	0.10	0.10	0.02	0.05

For each refinement archive, a header file, which records the lengths of the three refinement components, is attached to the refinement information to form a complete refinement archive package.

3.4 The Refinement Archive Decoding

In the receiver end, a base mesh is received followed by a sequence of refinement archives. The decoder first decodes the base mesh by *TG*'s algorithm, and then refines the base mesh by the upcoming refinement archives. The decompression steps of each refinement archive to reconstruct a mesh from M_i to M_{i+1} are stated as follows:

1. Decode the header file to get the length information of three components: the split vertex information, the *GPR*s information, and the connectivity information.
2. Decode the split vertex information to figure out the split vertices in M_i. For split vertices, the following geometry and connectivity information are decoded for operating a series of *vertex-split* operations.
3. Decode the *GPR*s information. The coordinates of a split child vertex can be calculated by

$$v_c = r - v_p + \frac{2}{|N(v_p)|} \sum_{v_j \in N(v_p)} v_j \tag{3}$$

   where r is the decoded *GPR*; $N(v_p)$ is the set of v_p's neighboring vertices in M_i, and $|N(v_p)|$ is the size of $N(v_p)$. The coordinates of the split child vertices are then used in the next step to get the connection relationship of the next finer mesh M_{i+1}.
4. Decode the connectivity information. For each *vertex-split* pair, we operate a *vertex-split* operation by re-connecting v_p and v_c to their neighboring vertices. Firstly, we connect each neighboring vertex to either v_p or v_c, depending on which one is nearer. This operation will leave two quadrilateral holes. Secondly, if the decoded signal is *SS*, *SL*, *LS*, or *LL*, we simply triangulate these two left quadrilateral holes according to the decoded information. Otherwise, the decoded signals must be two integers, which are the offsets of the two co-vertices between the *SS* condition and the actual condition. We can easily connect v_p and v_c, to the correct neighboring vertices after the two co-vertices have been calculated.

4 Experimental Results and Conclusion

We have implemented the proposed compression algorithm on various kinds of triangle meshes. We used the *TG* single resolution scheme to encode the base mesh M_0 in all our experiments. In average, it takes about 1.5 bits/*v* for the connectivity information, 9 bits/*v* for geometry information with 8-bit quantization per coordinate, and 13.5 bits/*v* for geometry information with 12-bit quantization per coordinate.

Table 2 shows the detailed compression results of our scheme in every hierarchy for the *Bunny* model, quantized to 10 bits per coordinate. The original Bunny model has 35947 vertices and 69451 faces, and is simplified to a crude model, M_0, with 3022 vertices and 3601 faces. The "*C* bits" column is the sum of the split vertex and the connectivity information. The split vertex information costs about 2.3 bits/*v* in each hierarchy and the connectivity information costs the others. Table 3 compares the results of our algorithm with that of *TG* [15], *PM* [7], and *CPM* [10] methods. These three mechanisms are the-state-of-the-art schemes for single resolution, continuous progressive, and hierarchical progressive mesh compression, respectively. Our scheme is superior to *CPM* in both geometry and connectivity compression. From the nature of the independent set split-vertex selection algorithm, edge collapse operations in the same hierarchy will be distributed averagely around the surface of the simplified geometric model. Thus this algorithm is progressively multi-resolution. Fig. 8 shows the experimental results of the *Horse* model in different hierarchies.

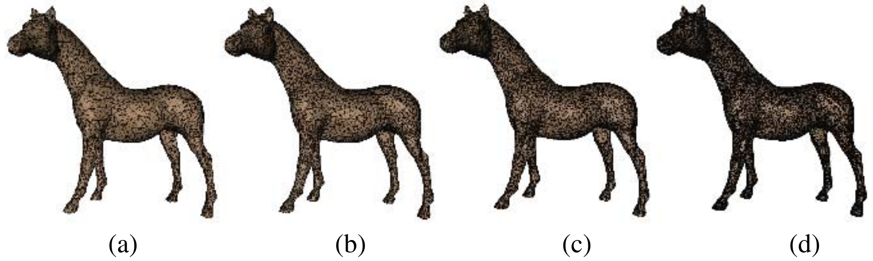


Fig. 8. The *Horse* model in different hierarchies: (a) The crude model, M_0, 2420 vertices, 4836 faces. (b) M_2, 4444 vertices, 8884 faces. (c) M_4, 8113 vertices, 16222 faces. (d) The finest model, M_7, 19851 vertices, 39698 faces.

Table 2. The detailed compression results of our scheme in each hierarchy for a *Bunny* model, quantized to 10 bits per coordinate. *G* represents "Geometry" and *C* represents "Connectivity".

Refinement Archive	vertices	*G* bits	*G*/*v*	Cumulative *G*/*v*	*C* bits	*C*/*v*	Cumulative *C*/*v*	Reconstructed Model
Base Model	3022	31006	10.26	--	3082	1.02	--	M_0
R_1	606	12130	20.02	20.02	3448	5.69	5.69	M_1
R_2	855	15537	18.17	18.94	4685	5.48	5.57	M_2
R_3	1125	18396	16.35	17.81	6142	5.46	5.52	M_3
R_4	1577	23579	14.95	16.73	8594	5.45	5.49	M_4
R_5	2083	28862	13.86	15.77	11185	5.37	5.45	M_5
R_6	2773	35694	12.87	14.88	14336	5.17	5.37	M_6
R_7	3668	41896	11.42	13.88	17899	4.88	5.22	M_7
R_8	4991	48083	9.63	12.68	21710	4.35	4.98	M_8
R_9	6793	64540	9.50	11.80	27987	4.12	4.74	M_9
R_{10}	8454	63760	7.54	10.71	27137	3.21	4.35	M_{10}

Table 3. Comparison of compression results (bits/*v*) of *TG*, *PM*, *CPM*, and our scheme. The geometry information is quantized to 10 bits per coordinate. The datum is the cumulative compression results for transmitting full resolution models.

		Single resolution		Continuous progressive		Hierarchy progressive			
Model	Vertex	*TG*		*PM*		*CPM*		Our scheme	
		G/*v*	*C*/*v*	*G*/*v*	*C*/*v*	*G*/*v*	*C*/*v*	*G*/*v*	*C*/*v*
Bunny	35947	*NA*	1.6	*NA*	37	15.4	7.2	10.71	4.35
Horse	19851	*NA*	1.4	*NA*	28	14.2	7.0	11.97	4.56
Triceratops	2832	10.4	2.2	*NA*	32	14.5	7.0	12.45	4.87

References

1. Bajaj, C.L., Pascucci, V., Zhuang, G.: Progressive Compression and Transmission of Arbitrary Triangular Meshes. In Proc. Visualization (1999) 307-316
2. Choi, J.S., Kim, Y.H., Lee, H.J., Park, I.S., Lee, M.H., Ahn, C.: Geometry Compression of 3-D Mesh Models using Predictive Two-Stage Quantization. IEEE Circuits and Systems for Video Technology, Vol. 10(2). (2000) 312-322
3. Cleary, J.G., Neal, R.M., Witten, I.H.: Arithmetic Coding for Data Compression. ACM Comm., Vol. 30(6). (1987) 520-540
4. Cohen-Or, D., Levin, D., Remez, O.: Progressive Compression of Arbitrary Triangular Meshes. In Proc. Visualization (1999) 62-72
5. Deering, M.: Geometry Compression. In Proc. SIGGRAPH (1995) 13-20
6. Heckbert, P.S., Garland, M.: Survey of Polygonal Surface Simplification Algorithms. In Proc. SIGGRAPH Course Notes 25. (1997)
7. Hoppe, H.: Progressive Meshes. In Proc. SIGGRAPH (1996) 99-108
8. Hoppe, H.: Efficient Implementation of Progressive Meshes. Technical Report MSR-TR-98-02. Microsoft Research. (1998)
9. Li, J., Kuo, C.C.: Progressive Coding of 3D Graphics Models. In Proc. IEEE. Vol. 96(6). (1998) 1052-1063
10. Pajarola, R., Rossignac, J.: Compressed Progressive Meshes. IEEE Visualization and Computer Graphics, Vol. 6(1). (2000) 79-93
11. Peng, J., Kim, C.-S., Kuo, C.-C. J.: Technologies for 3D Mesh Compression: A Survey. J. Vis. Commun. Image, R. 16. (2005) 688–733
12. Rossignac, J.: Edgebreaker: Compressing the Incidence Graph of Triangle Meshes. IEEE Visualization and Computer Graphics, Vol. 5(1). (1999) 47-61
13. Soucy, M., Laurendeau, D.: Multiresolution Surface Modeling Based on Hierarchical Triangulation. Computer Vision and Image Understanding, Vol. 63. (1996) 1-14
14. Taubin, G., Rossignac, J.: Geometric Compression Through Topological Surgery. ACM Graphics, Vol. 17(2). (1998) 84-115
15. Touma, C., Gotsman, C.: Triangle Mesh Compression. In Proc. Graphics Interface (1998) 26-34
16. Valette, S., Prost, R.: A Wavelet-Based Progressive Compression Scheme for Triangle Meshes: Wavemesh. IEEE Visualization and Computer Graphics, Vol. 10(2). (2004) |123-129

Shape Preserving Hierarchical Triangular Mesh for Motion Estimation

Dong-Gyu Lee[1] and Su-Young Han[2]

[1] Dept. of Computer Engineering, Hanbuk University,
Sangpae-dong, Dongducheon, Kyounggi-do, 483-120, Korea
dglee@hanbuk.ac.kr
[2] Dept. Of Computer Science, Anyang University,
Samseong-ri, Buleun-myeon, Ganghwa-gun, Incheon, 417-833, Korea
syhan@anyang.ac.kr

Abstract. In this paper, we propose a motion estimation method using hierarchical triangulation that changes the triangular mesh structure according to its motion activity. The subdivision of triangular mesh is performed from the amount of motion that is calculated from the variance of frame difference. As a result, node distribution is concentrated on the region of high activity. The subdivision method that makes it possible to yield hierarchical triangular mesh is proposed as well as the coding method reducing additional information for hierarchical mesh structure is described. From the experimental result, the proposed method has better performance than the conventional BMA and the other mesh based methods.

Keywords: Motion Estimation, Motion Compensation, Triangular Mesh.

1 Introduction

Block-based motion estimation and motion compensation is widely employed in modern video-compression systems. The block match algorithm(BMA) assumes that the object displacement is constant within a small block of pixels. This assumption presents difficulties in scenes with multiple moving objects and rotating objects. When adjacent blocks are assigned different displacements, a BMA may form a discontinuity in the reconstructed image called blocking artifacts.

Triangular and quadrilateral meshes were proposed as an alternative to BMA in order to allow for affine or bilinear motion estimation and thus achieve better performance in the complex motion field.

There are two types of mesh structure, regular meshes [1-2] and content-based meshes [3-5]. In regular meshes, nodes are placed at fixed intervals throughout the image. It is not need to send the mesh topology to the decoder. But this may not have the ability to reflect scene content, i.e., a single mesh element may contain multiple motion. Methods using content-based meshes address this problem, which extract the scene characteristic points and apply triangulation to the points. However these methods require much overhead information about the structure of mesh and the position of nodes.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 929 – 938, 2006.

In this paper, we propose a hierarchical triangular mesh generation method, in which patches that have high motion activity are successively subdivided, and the additional information reduction coding method for hierarchical triangular mesh structure.

2 Hierarchical Triangular Mesh (HTM)

The first step of mesh-based methods is to divide the image into many triangular patches. The motion vectors of the triangular patches are estimated by the node displacement and then the moving position of node is obtained. The motion vectors of the three vertices of a triangular patch are employed in determining the six parameters of an affine transform. In backward motion estimation, the displacement of every pixel inside the triangle can also be obtained from the previous frame using the affine transformation.

In generating hierarchical mesh structure, we use the regular meshes which are isosceles right triangle in the coarsest level (level 0). The nodes in this mesh have four-way or eight-way connectivity. According to the motion activity, a new node is introduced in the region which has large motion to construct a finer triangle.

The variance of the frame difference (FD) between the previous frame and the current frame is chosen as a measuring function to judge whether the region should be subdivided or not (see Equation (1)).

$$m(P) = \frac{1}{N_p} \sum_{(x,y)\in p} (f_{FD}(x,y) - \overline{f_{FD}})^2 \tag{1}$$

Where the region P is the triangular patch region, N_P is the number of pixels in the region, $f_{FD}(x,y)$ is the frame difference between current frame and previous frame and $\overline{f_{FD}}$ is the mean of $f_{FD}(x,y)$. If $m(P) > TH$, the three nodes are added into the sides of the triangular respectively. Otherwise, the node is not added because of the small motion value.

Because adjacent triangle meshes share a side, the introduced nodes change the structure of adjacent meshes. From the node addition, the next cases are occurred.

(1) If all of the current mesh and adjacent meshes are not influenced by node addition, the number of nodes on the sides of the triangle is 0.
(2) If the node is added to the adjacent meshes and not added to the current mesh, the number of nodes on the sides of the triangle is 1, 2 (see Fig.1(a)-(e)).
(3) If the node is added to the current mesh or is added to all adjacent meshes, the number of nodes on the sides of the triangle is 3 (see Fig.1(e)-(h)).

Figure 1 illustrates the possible triangulation according to the number and the position of the nodes.

In the proposed algorithm, for consecutive subdividing a triangle, the shape of a mesh in a coarse level must be preserved in a finer level, too. Mesh structures in all levels are restricted to isosceles right triangle. According to the number and the position of nodes on the sides, the algorithm of triangulation is as follows.

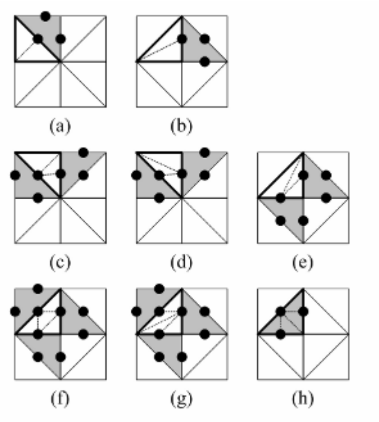


Fig. 1. The possible triangulation according to the number and the position of the nodes

Case 0. *If the constructed mesh is already an isosceles right triangle, more process is not necessary (see Fig. 1(a),(c),(f),(h)).*
Case 1. *If all nodes are located at only the legs of isosceles right triangle, a new node is introduced on hypotenuse of isosceles right triangle (see Fig. 1(e)).*

Figure 2 represents the method of isosceles right triangulation. Here the vector $\vec{a}$ is from a node on side to the opposite vertex and the vector $\vec{b}$ is a side vector containing the node. For generated triangle to be isosceles right triangle, it is necessary that vector $\vec{a}$ and $\vec{b}$ for all nodes are orthogonal. If not, a node must be introduced on the hypotenuse of this triangle.

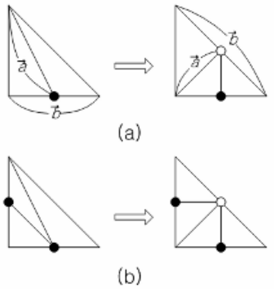


Fig. 2. Isosceles right triangulation by node addition

The proposed triangulation procedure is as follows.

Step 1) In level 0, generate regular meshes, which are isosceles right triangles.
Step 2) Examine whether the distance between triangle vertexes is minimum distance on that level and determine whether patch is split or not. If the amount of motion activity is larger than given threshold, add node into all sides of triangle.

If $m(P) > TH$ *, node is introduced into the all sides.*

Step 3) In order for all meshes to be isosceles right triangles, evaluate the inner product of the side vector $\vec{b}$ containing the node and the vector $\vec{a}$ from the node to opposite vertex. If it is not orthogonal, add a new node to hypotenuse. Iterate the procedure over the all meshes until adding a node is not necessary.

If $\vec{a}\bullet\vec{b} = 0$, *node is introduced into the hypotenuse.*

Step 4) Triangulate the patches according to the number of nodes on each side of triangle.
Step 5) Increase a level by one and divide the minimum size of triangle by two. Repeat Step 2)

Fig. 3 indicates this triangulation procedure and Fig. 4 shows the result of sample image.

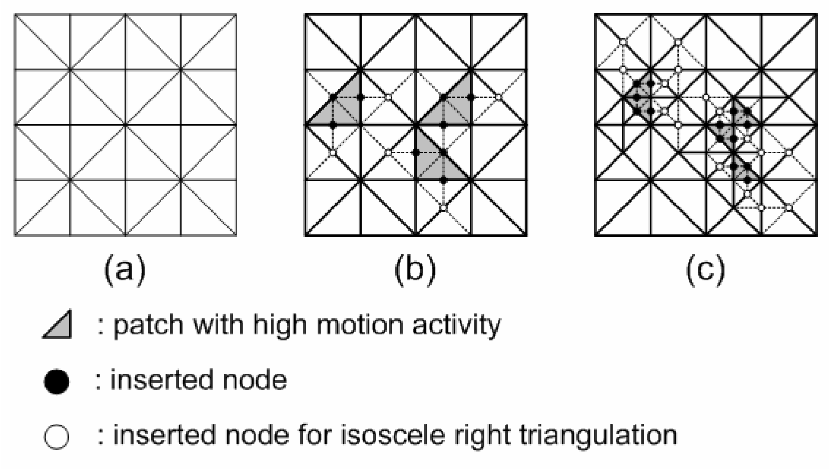
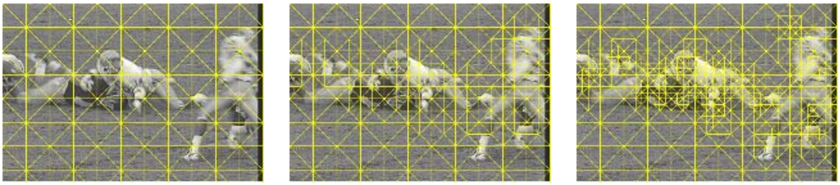

(a)
(b)
(c)
◺ : patch with high motion activity
● : inserted node
○ : inserted node for isoscele right triangulation

Fig. 3. The example of hierarchical triangulation (a) level 0 (b) level 1 (c) level 2



(a) (b) (c)

Fig. 4. Hierarchical triangular mesh(Football-frame No.2) (a) level 0 (b) level 1 (c) level 2

3 Estimation of Motion Vector in HTM

In order to estimate the affine transformation parameters for motion compensation, it is required to estimate the motion vectors at nodes. The motion vectors of the three vertices of a triangular patch are employed in determining the six parameters of an affine transformation. We use the BMA to get the approximate displacement and then apply the hexagonal matching algorithm(HMA) as a refinement. In order to estimate the moving position of a node, we refine a considered node at the position, where the difference between the reconstructed and the original images is minimal

3.1 Node Motion Estimation Using BMA

BMA centering a considered node is used to search the approximate displacement of node. Because in hierarchical mesh, the distance between nodes is too short as level increases and the additional refinement process is introduced, BMA is applied to the nodes in only level 0. For the other nodes, bilinear interpolation using the position of nodes in level 0 is used to estimate the node position.

3.2 Affine Motion Compensation

After the displacement of the tree vertices of a triangular patch is obtained, the image is reconstructed using the affine transformation. For each point (x, y) of a considered triangle, the corresponding transformed position (x', y') is given by

$$\begin{aligned} u(x,y) &= (x-x') = a_1 + a_2 x + a_3 y \\ v(x,y) &= (y-y') = a_4 + a_5 x + a_6 y \end{aligned} \tag{2}$$

where $a_1,\cdots,a_6$ coefficients are the six motion parameters of the considered triangle. Let us consider that Equation (2) is available in each vertex of triangle:

$$\begin{bmatrix} u(x_1,y_1) \\ u(x_2,y_2) \\ u(x_3,y_3) \\ v(x_1,y_1) \\ v(x_2,y_2) \\ v(x_3,y_3) \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ 1 & x_2 & y_2 & 0 & 0 & 0 \\ 1 & x_3 & y_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ 0 & 0 & 0 & 1 & x_2 & y_2 \\ 0 & 0 & 0 & 1 & x_3 & y_3 \end{bmatrix} = \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix} a \tag{3}$$

where $\Phi = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}$ and by rewriting the vector a as a function of Φ^{-1}, Equation (3) becomes

$$a = \begin{bmatrix} \Phi^{-1} & 0 \\ 0 & \Phi^{-1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{4}$$

where

$$\Phi^{-1} = \frac{1}{\det(\Phi)} \times \begin{bmatrix} x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_2 - x_2y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix} \tag{5}$$

By substituting the affine parameters into Equation (2), we can perform motion compensation of any point in the current triangle. Because the affine parameters are non-integer real numbers, the resulting coordinates in the reference frame are too. In this case, bilinear interpolation is used to estimate the pixel location.

3.3 Node Refinement

In order to estimate the moving position of a node, we refine a considered node at the position, where the difference between the reconstructed and original images is minimal. Because moving a node affects the structure of adjacent meshes, while keeping neighboring nodes fixed, we move a considered node and reconstruct the image by affine transformation. Motion vector is decided from the position that minimizes the difference. PSNR as the cost function is defined such as

$$\begin{aligned} PSNR &= 10\log_{10}\frac{255^2}{MSE}dB \\ MSE &= \frac{1}{N_P}\sum_{(x,y\in P)}(I(x,y)-\overline{I(x,y)})^2 \end{aligned} \tag{6}$$

where P is the region bounded on neighboring nodes and N_P is the number of pixels in that region and $I(x,y)$, $\overline{I(x,y)}$ is the pixel intensity of the original image and the reconstructed image, respectively.

The refinement process is done on a finer level after all nodes in a coarse level were refined. The above refinement process is iterated until the positions of nodes are optimized.

If neighboring nodes were not moved on the previous refinement process and a considered node is not moved on the current refinement process, the refinement process is finished.

4 Coding of the Hierarchical Triangular Mesh

Contrast to BMA or regular mesh, the proposed mesh structure is deformable, so additional information for describing the mesh structure is required. In this paper, because the subdivision of triangular mesh is based on the motion activity, additional information can be minimized from coding the only existence of added node. The reconstruction of the mesh structure is accomplished by the same mesh generation process according to the subdivision information.

We let code ‘0’ represent the number of node addition is one or zero, and code ‘10’ represents the number of node addition is two, ‘11’ represents that the number of node addition is three, that is, ‘1’ represents the patch that can be subdivided in the higher

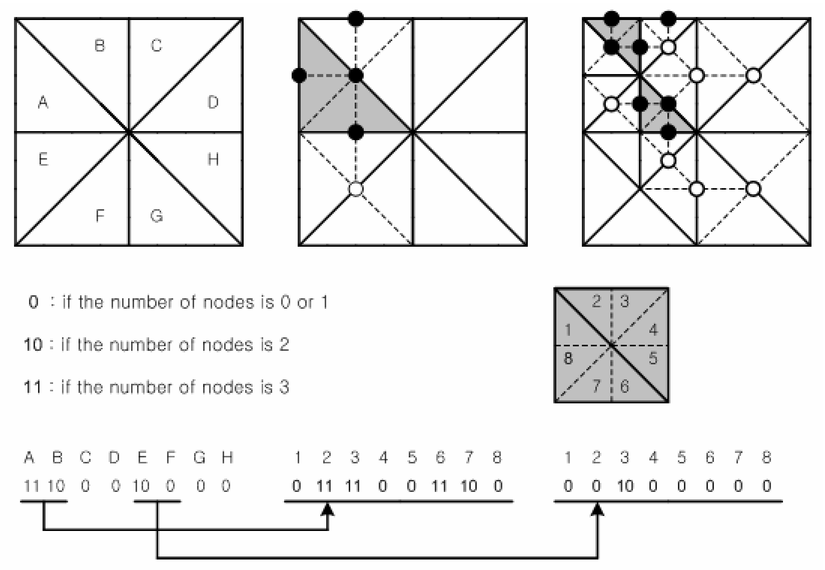


Fig. 5. The coding of hierarchical mesh structure

composed of adjacent two triangle is subdivide into 8 patches, and the number of added node is coded for each patch.

For the reconstruction of the mesh structure, if ‘11’ code is encountered in the received stream, three node is added to the patch and new node is connected with each other to make an isosceles right triangle.

5 Experimental Results

The performance of the proposed mesh structure was evaluated for two different test sequences, using the various mesh structure. These sequences(Clair, Football) consist of 352×288 pixels and 352×240 pixels respectively. Clair is tested at 3 frame interval for the sequence number 1 to 100 and Football is tested at every frame for the sequence number 1 to 30.

In the simulation, the performance of the proposed hierarchical triangular mesh(HTM) is compared with that of BMA, hexagonal matching algorithm(HMA) [1] and hierarchical grid interpolation(HGI) [6]. Motion vector is estimated for each node and then from the displacement of these nodes, the reconstructed image is calculated by the motion compensation. As a measure of the quality of the reconstructed image, the PSNR between the original image and the reconstructed image was used.

We use full-search BMA with search region of [-8, +7] pixel both in the horizontal and vertical direction respectively. In regular mesh, node distance is 16×16 pixel and initial node search region with BMA is [-8, +7] and search region in the iterative refinement process was [-3, +3]. The other conditions are the same as those of regular mesh.

We used the 3-level hierarchical structure in HTM. And minimum node distance is 32×32 in the level 0 and 16×16 in the level 1 and 8×8 in the level 2. For each triangular patch in which the variance of frame difference is larger than the threshold, additional nodes are inserted to each side. Node refinement is processed from low level nodes to higher level nodes. For the sake of performance comparison with the other mesh based methods, we adjusted the number of nodes in HGI and HTM similar to those of the other methods. The threshold value for node addition is renewed from the follow Equation (7), which makes total node the same as in the regular mesh, that is about 437 ± 5 %

$$TH_{\mathrm{var}}(n+1)=TH_{\mathrm{var}}(n)\times\left\{1+\frac{(TotalNode-437)}{437}\right\} \tag{7}$$

where $TH_{\mathrm{var}}(n)$ and $TH_{\mathrm{var}}(n+1)$ are the variance threshold of the current iteration and the next iteration respectively, $TotalNode$ is the number of total nodes.

Table 1. Average PSNR (dB)

Input Image	Algorithm	After Complete Convergence	After 3 Iteration
Clair	BMA	36.0098	
	HMA	38.0738	37.9504
	HGI	38.6513	38.4131
	HTM	39.1313	38.9704
Football	BMA	22.7097	
	HMA	23.1569	22.3779
	HGI	23.5748	22.7508
	HTM	24.1643	23.9737

During the node refinement or image reconstruction, the pixel value was calculated from the bilinear interpolation which was used for subpixel position produced by affine transform. In general, two or three iterative node refinements are sufficient in the refinement process. For the performance comparison, Fig. 6 and Fig. 7 represent the PSNR of reconstructed image in case of after complete convergence and Table 1 shows the average PSNR for the each condition.

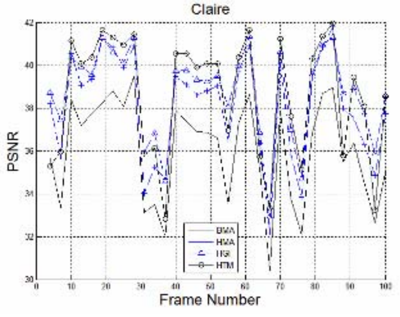


Fig. 6. The PSNR of motion compensated image (Claire)

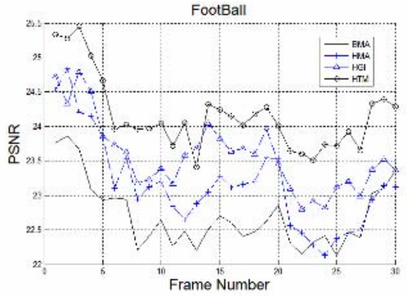


Fig. 7. The PSNR of motion compensated image (Football)

Table 2 shows the total number of nodes and the average bitrate for coding the mesh structure. Each motion vector is allocated with 4 bits for horizontal and vertical direction respectively. The total transmitted bits in HGI and HTM are the sum of the bits for coding the mesh structure and the bits for motion vector. In this case because the number of nodes is variable, the average bits per image are presented.

In Fig. 6 and Fig. 7, we can see that HTM achieved the better performance in PSNR compared with the regular mesh or HGI with the similar number of nodes. Contrary to regular mesh, HTM structure requires the overhead information for mesh structure coding. But using the appropriate coding method, the mesh structure can be efficiently encoded.

The proposed method is superior to regular mesh in average PSNR about 0.5dB to 1 dB. Especially, higher performance is achieved in case of large motion activity image. In Table 1, the average PSNR after the three iterations are shown, this result represents that proposed algorithm improves the node convergence speed in refinement process compared with other mesh based method.

Table 2. Average Bitrate (Transmitted)

Input Image	Algorithm	Average Bits
Clair	BMA	396*8
	HMA	428*8
	HGI	430.6*8+452.21
	HTM	434.6*8+666.27
Football	BMA	330*8
	HMA	362*8
	HGI	437.5*8+461.5
	HTM	439.0*8+718.1

6 Conclusion

In this paper, a new hierarchical mesh generation method has been proposed. The size of triangular patch is varied according to motion activity of a video sequence. Because the mesh generated is denser in moving area than in still area, the motion compensation pays more attention on moving area such that better performance is achieved. We exploit the hierarchical mesh structure that the subdivided patches construct the isosceles right triangle like the initial triangular patch. It makes the additional subdivision is possible and reduces the overhead information for coding the mesh structure.

Simulation result shows that the proposed algorithm improves the performance of motion compensation in comparison with the other mesh based methods by more than 0.5dB or 1 dB in PSNR and is also suitable for low bitrate coding.

References

[1] Y. Nakaya and H. Harashima: Motion Compensation Based on Spatial Transformations. IEEE Trans. on Circuits and Systems for Video Technology, 4(3), June (1994) 339-356

[2] M. Sayed and W. Badawy: A Novel Motion Estimation Method for Mesh-Based Video Motion Tracking. ICASSP 2004, (2004) 337-340

[3] M. Dundon, O. Avaro, C. Roux: Triangular active mesh for motion estimation. Signal Processing: Image Communication, 10, (1997) 21-41

[4] G. Al-Regib, Y. Altunbasak and R. M. Mersereau: Hierarchical Motion Estimation with Content-Based Meshes. IEEE Trans. On Circuits and Systems for Video Technology, 13(10), Oct. (2003) 1000-1005

[5] M. Servais, T. Vlachos and T. Davies: Affine Motion Compensation using a Content-Based Mesh. IEE Proc.-Vis. Image Signal Process., 152(4), August (2005) 415-423

[6] C-L Hung and C-Y Hsu: A New Motion Compensation Method for Image Sequence Coding Using Hierarchical Grid Interpolation. IEEE Trans. on Circuits and Systems for Video Technology, 4(1), Feb. (1994) 42-52

Parameter Selection of Robust Fine Granularity Scalable Video Coding over MPEG-4

Shao-Yu Wang and Sheng-Jyh Wang

National Chiao Tung University
1001 Ta Hsueh Road, Hsin Chu, Taiwan, R.O.C.
ponpon.ee88@nctu.edu.tw, shengjyh@cc.nctu.edu.tw

Abstract. In this paper, we propose an approach to automatically choose the α and β parameters in the RFGS coding scheme for the coding of video sequences. We first analyze the residuals after motion compensation with different parameter settings under various bit rates. We then investigate the relationship between the MSE gain and MSE loss for each bit-plane. With those models and relationships, we propose first an approach to automatically choose the sets of optimized parameters for a fixed bit rate. Then we propose an approach to deal with a range of bit rates. These two MSE-based approaches require only light computational loads.

1 Introduction

In the past decades, video compression has drawn significant attention due to the challenging task to transmit video data through channels with limited bandwidth. Traditionally, video compression standards, like MPEG-1, aims at optimizing the coding efficiency at a fixed bit rate. To transmit through wireless networks, however, video data had better be coded in a scalable way since the allowed bandwidth is dynamically varied. Therefore, for wireless transmission, video sequences need to be optimized for a wide range of bit rates, instead of a fixed bit rate. In MPEG-4, the fine granularity scalability (FGS) scheme has been proposed to offer a scalable way to cope with bandwidth fluctuation [1]. The concept of FGS is illustrated in Fig. 1. In FGS, a video sequence is partitioned into two layers: a base layer which satisfies the minimum bandwidth requirement, and an enhancement layer which is obtained by subtracting the base layer from the original sequence. The base layer is coded using DCT-based non-scalable single layer coding, while the enhancement layer is coded using bit-plane coding and can be truncated at any bit [2]. In the FGS scheme, the motion compensation process is only applied over the base layer and the enhancement layer is coded without temporal predictions. Although this scheme may enhance the capability of error resilience, the lack of temporal prediction in the enhancement layer may sacrifice the coding efficiency.

After the FGS proposal, several approaches have been proposed to increase the coding efficiency by exploiting the temporal correlation [3],[5]-[8]. Among these approaches, the robust fine granularity scalability (RFGS) scheme proposed in [3] offers a flexible structure to control the level of temporal prediction. In RFGS, two parameters, α and β, are utilized to control the temporal prediction process, as shown

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 939 – 948, 2006.

in Fig. 2. The parameter β, with $0 \le \beta \le N_b$, controls the number of bit-planes in the enhancement layer that would be used for motion compensation; while the parameter α, with $0 \le \alpha \le 1$, is a leaky factor that can be used to attenuate the amount of error drift. Here, N_b denotes the number of bit-planes in the enhancement layer. For more details, please refer to [3].

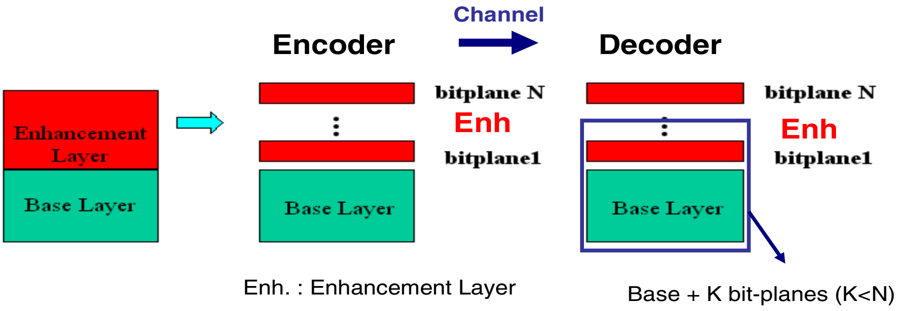


Fig. 1. Enhancement Layer and Base Layer in FGS

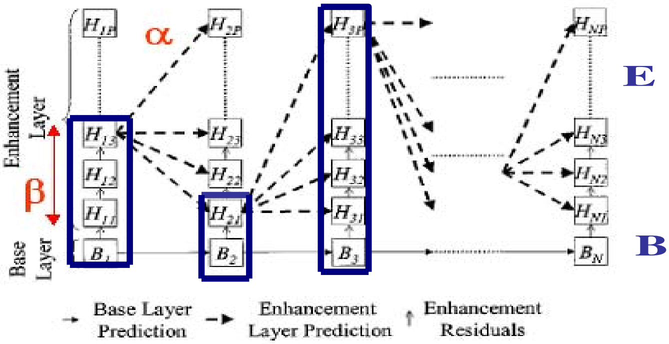


Fig. 2. Illustration of the RFGS scheme [3]

Since scalable coding focuses on optimizing coding efficiency over a given range of bit rates, we need a mechanism that holds for the entire range of interest. The temporal prediction in the enhancement layer, however, may produce different effects at different bit rates. If a higher bit rate is allowed, a more detailed reference frame can be used to increase the efficiency of motion compensation. On the contrary, in a lower bit-rate scenario, a less detailed reference frame would be preferred for motion compensation due to the possible error drifts caused by loss of the reference frame data during the transmission. In this paper, we investigate the interaction between motion compensation and error drifts, and try to optimize the amount of temporal prediction over a given range of bit rates. Here, we propose an approach to adaptively select the parameters α and β in the RFGS scheme. The rest of this paper is organized as follows. In Section II, the properties of different videos are analyzed to set up some appropriate models and relationships. Based on these models and relationships, two approaches are proposed in Section III. Finally, in Section IV, we conclude this paper.

2 Problem Modeling

In this section, the characteristics of residual will be studied. The reason why we choose MSE, instead of PSNR, as the base of discussion will be briefly introduced. Based on MSE, we will analyze the gain and loss of video quality based on the MSE measure. Here, we'll discuss the reduction of MSE in the decoded frame when the encoded data are successfully received and the increase of MSE when the encoded data are lost during the transmission.

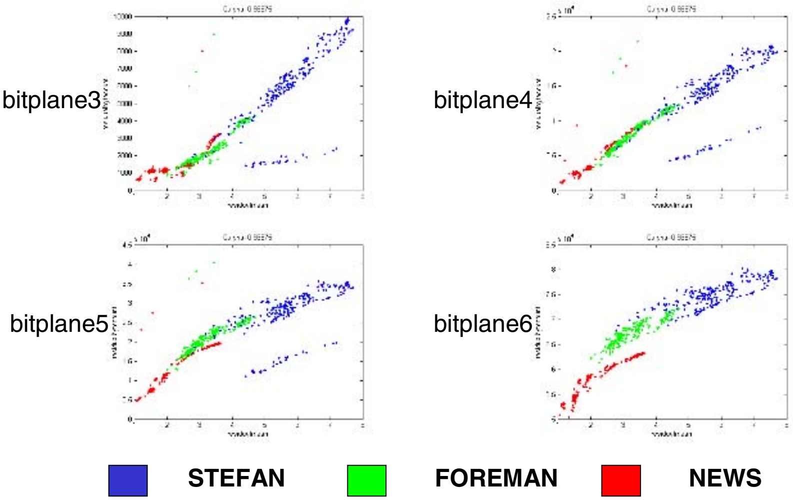


Fig. 3. Encoded byte counts v.s. residual mean of a bitplane

2.1 Residuals

A main work of the encoder is to encode the residuals of motion compensation, which cannot be compensated by temporal prediction, into the output bit-stream. At an encoder, motion estimation and motion compensation are executed first. Then, after quantization and entropy coding, we get the final bit-stream. In the bit-plane coding of the RFGS scheme, residuals are coded into several bit-planes. With varied parameter α and β, the encoded bit-streams could be rather distinct. In our experiments, we first observed for each bit-plane the relationship between the encoded byte count and the mean of the residual at that bit-plane. As shown in Fig. 3 and Table 1, it was found that the relationship is close to linear for various video sequences and for different settings of α and β. Hence, according to the rules established in Table 1, we could get a rough prediction of the encoded byte count based on the residual of the bitplane after the motion compensation process.

For the 1st bit-plane and the 2nd bit-plane, on the other hand, the byte counts are usually small. For these two bit-planes, the linear model may be deficient. Fortunately, unless the network condition is too bad, the data of these two bit-planes are usually successfully transmitted.

Table 1. Modeling of the relation between residula mean and the encoded byte count

Bit-plane	Number of Encoded Bytes
Bit-plane3	≈ M * 750 + 500
Bit-plane4	≈ M * 2250 + 1000
Bit-plane5	≈ M * 3750 + 5000
Bit-plane6	≈ M * 3750 + 50000

Remark: M = Residual Mean of the Bitplane

2.2 MSE vs. PSNR

In the literature, both PSNR and MSE are widely used as the measures to represent video quality. As expressed in Equation (1), the relation between PSNR and MSE is not linear. At high PSNR, a little decrease of MSE may bring up a considerable improvement of PSNR; while at low PSNR, PSNR grows up slowly with the decrease of MSE. As to be explained in the following subsections, the MSE measure may offer simple and straightforward ways to select α and β. Hence, in the proposed approach, the MSE is chosen as the reference measure for the selection of α and β.

$$PSNR = 20 \times \log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{1}$$

2.3 Modeling of MSE Reduction

With a larger value of α and β, a more efficient motion compensation is expected. As long as the reference frame can be fully transmitted to the decoder site, the visual quality is expected to be better under the same bandwidth condition. For the residual part, as more bytes are received at the decoder, the MSE of the decoded frame is expected to be lower. In our simulation, we intentionally control the exact byte counts transferred to the decoder site and observe the relation between the transmitted byte counts and the MSE value of the decoded frame. To simplify the problem, we fix α and only allow β to vary. As shown in Fig.4, it is found that the relationship between the MSE value and the transmitted byte count is basically piecewise-linear. Each line segment corresponds to the transmission of one bit-plane. This linear property indicates that, for each bit-plane, every received byte has roughly the same contribution to the reduction of the MSE value in the decoded frame. Fig. 5 shows the relationship between the slopes of the line segments and the MSE value. The slope indicates the decreased MSE in the decode frame when every 200 bytes are transmitted to the decoder site. By summarizing Fig. 5 and Fig. 6, we can deduce Table 2, which lists the rough estimation about the reduction of MSE in the decoded frame for every 200 received bytes at the decoder site. Hence, based on the available bandwidth and the byte counts in various bit-planes, the encoder will be able to approximately estimate the expected visual quality at the decoder site.

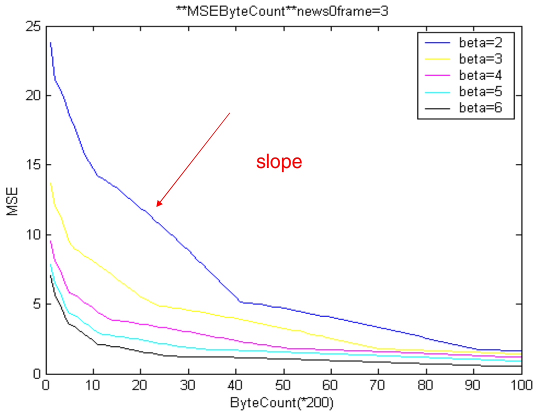


Fig. 4. MSE vs. received byte count

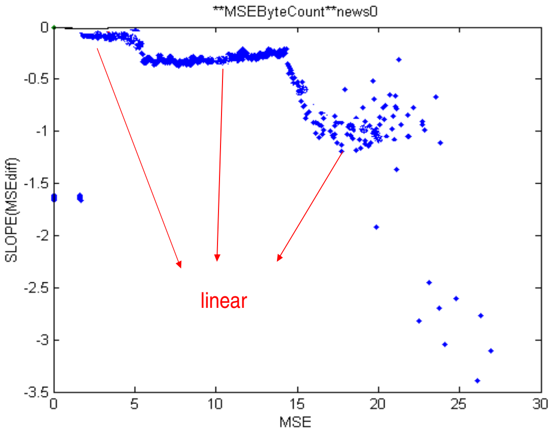


Fig. 5. Decreasing rate of MSE vs. MSE

Table 2. Relationship between the decreased MSE and the received byte count

Bit-plane	reduced MSE per 200 received bytes
Bit-plane 2	3.5
Bit-plane 3	1
Bit-plane 4	0.3
Bit-plane 5	0.1
Bit-plane 6	0.05

2.4 Modeling of Error Drifts

On the other hand, if the network bandwidth is not wide enough to fully transmit the reference frame to the decoder site, then the motion compensation process will produce errors that drift in the subsequent frames. To model the effect of error drift, we perform another simulation as follows.

First, we individually decrease the bit-rate at the first frame and the second frame and measure the corresponding increase of MSE at the third frame. Then we decrease the bit-rates at the first and the second frames simultaneously and measure the overall increase of MSE at the third frame. The results are shown in Fig. 6. It is found that the superposition of the individual increase of MSE (purple curve) is roughly equal to the overall increase of MSE (cyan curve). Hence, in the following discussions, we first investigate the increase of MSE caused by the data loss in a single frame. Then we accumulate the overall increase of MSE based on this superposition property.

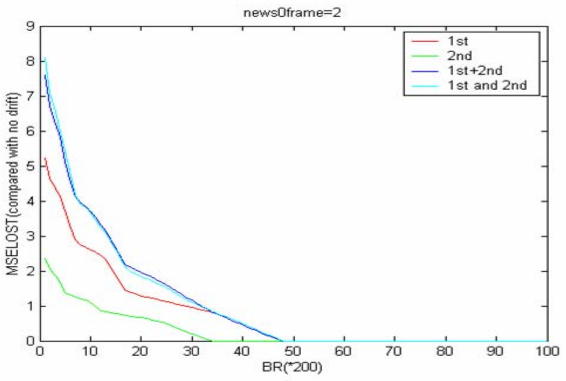


Fig. 6. Increased MSE versus the lost data in the reference frames

In RFGS, due to the use of the parameter α, error drifts will decrease with a geometric ratio in the consecutive frames [3]. With a fixed β, we simulate the decay of error drifts caused by the data loss at a single reference frame with respect to different α's. In Fig. 7, the simulation result shows that the decay of MSE values indeed follows the geometric behavior.

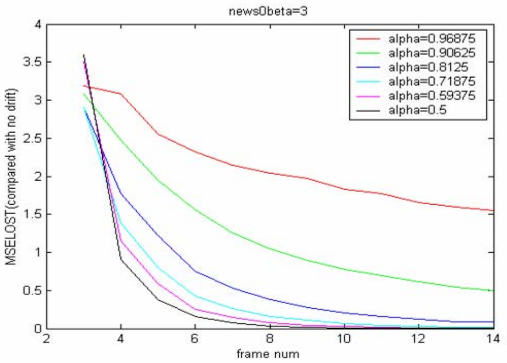


Fig. 7. Decay of error drifts with respect to different α's

Based on the above simulations, for the data loss at the current frame, we can model the overall increased MSE generated in the subsequent frames to be the product of the increased MSE at the next frame and the factor $(1+\alpha+\alpha^2+\alpha^3+\cdots)$. That is, we have

$$\textbf{Total increased MSE} \\ = \text{“}\textbf{Increased MSE at the next frame}\text{”} * (\,\mathbf{1} - \boldsymbol{\alpha}^{\mathbf{n}}) \,/\, (\,\mathbf{1} - \boldsymbol{\alpha}\,) \tag{2}$$

To model the increased MSE at the next frame, we perform simulations to observe the change of MSE at the next frame versus the byte count at the current frame. The relations are shown in Fig. 8. Similar to the discussion in the previous subsection, the curves are piecewise linear and each line segment corresponds to one bit-plane. Hence, we can deduce Table 3, which models the relationship between the increased MSE at the next frame and the lost byte counts at a bit-plane of the current frame. With Table 3 and Equation (2), we can then roughly estimate the overall impact of error drifts.

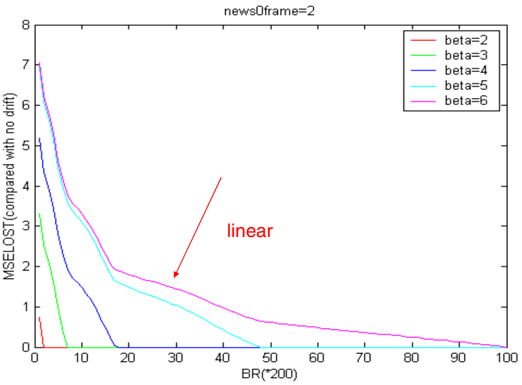


Fig. 8. The change of MSE at the next frame versus the byte count at the current frame

Table 3. Relation between the increased MSE at the next frame versus the lost byte count at a bit-plane of the current frame

Bit-plane	increased MSE per 200 lost bytes
Bit-plane 2	3.5
Bit-plane 3	1
Bit-plane 4	0.3
Bit-plane 5	0.1
Bit-plane 6	0.05

3 Proposed Approaches and Experimental Results

According to the rules and the models mentioned in Section 2, we can roughly estimate the size of the encoded bitstream right after testing a few choices of β's for motion compensation. That is, based on the residual mean at a bit-plane, we can roughly estimate the encoded byte count for that bit-plane. Then, based on the byte count of each bit-plane, we can estimate the reduced MSE in the decoded frame assuming that the encoded bytes can be successfully transmitted to the decoder. On the other hand, we may also estimate the increased overall MSE in the subsequent frames once if the encoded bytes are lost during the transmission. In the following subsections, two approaches for the selection of α and β will be proposed based on the aforementioned estimations.

3.1 Proposed Approach for a Fixed Bit Rate

As aforementioned, the parameter β determines the number of bit-planes that would be included in the reference frame for motion compensation. If all these β bit-planes can be successfully transmitted to the decoder, then a larger β is expected to be more efficient. Moreover, for each bit-plane, we may use the model mentioned in Section 2.3 to estimate the expected reduction in MSE. On the other hand, if the bit-planes in the reference frame cannot be transmitted to the decoder site, then a larger β may cause more serious error drifts. Based on the model mentioned in Section 2.4, we can estimate the increased MSE caused by error drifts.

In Fig. 9, we illustrate the proposed approach when the channel bandwidth is fixed. Here, based on the residual at each bit-plane, we can estimate the encoded byte count. Based on the channel bandwidth and the estimated byte count for each bit-plane, we can roughly estimate which bit-planes can be successfully transmitted, which bit-plane can only be partially transmitted, and which bit-planes cannot be transmitted. For example, assume Bit-planes 1~3 are expected to be available at the decoder site, Bit-plane 4 is expected to be partially available, and Bit-planes 5~6 are expected to be lost during the transmission. Then, we estimate the MSE gain (the reduced MSE in the decoded frame) and MSE loss (the increased MSE due to error drifts) for Bit-plane 4. If MSE gain is larger than MSE loss, then we choose β = 4 for this example. Otherwise, we choose β = 3.

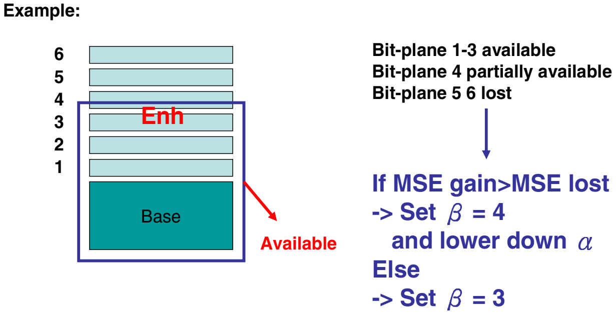


Fig. 9. Illustration of the proposed approach for a fixed bit-rate

On the other hand, the parameter α is to be chosen depending on whether there is error penalty or not. If the reference image for motion compensation can be fully received at the decoder under the channel bandwidth, we set α to be the upper bound

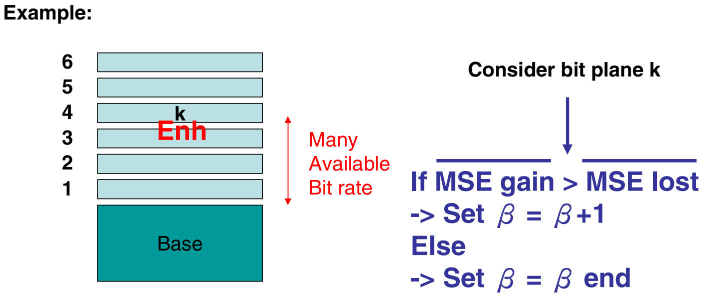


Fig. 10. Illustration of the proposed approach for a range of bit-rates

0.96875. Otherwise, once if error drifts are expected, we will reduce the value of α by one predefined step size for the subsequent 5 frames.

3.2 Proposed Approach I for a Range of Bit Rates

In real networks, the bandwidth may be varying. For this case, we choose several bit-rates within the range of variation as the checking points. At each check point, we apply the approach proposed in Section 3.1. Then, for each frame, we estimate the average of MSE gain and the average of MSE loss bit-plane by bit-plane, starting from Bit-plane 1. For the current bit-plane, if the averaged MSE gain is larger than the averaged MSE loss, it means the inclusion of this bit-plane in motion compensation may lower down the averaged MSE. Hence, we may increase β by 1 in this case. Otherwise, the current bit-plane will not be included in motion compensation. An illustration of this approach is shown in Fig. 10.

On the other hand, we choose the parameter α as follows. If error drifts are expected for more than a quarter but less than one half of the checking points under the selected β, then we lower down α by one step size in the subsequent 5 frames. On the other hand, if more than one half of the checking points suffer from error drifts, then we lower down α by two step sizes in the following 5 frames.

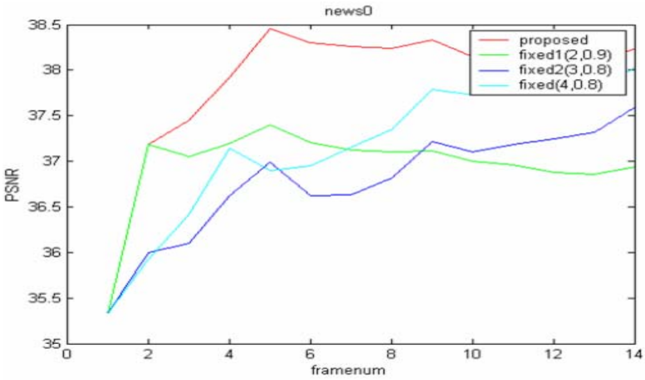


Fig. 11. Experimental results of the proposed approach

In Fig. 11, we show an experimental result. Here, we choose the checking points to be at 768K, 896K, 1024K, 1280K, 1400K, 1536K, 1750K, and 2048K. It can be seen that the adaptive parameter selection based on the proposed approach may offer better performance if compared with the cases with a fixed selection of parameters.

4 Conclusions

In this paper, we study the selection of the parameters α and β in the RFGS coding scheme. By statistics, several models and rules are established. Based on the models and rules, two fast algorithms are proposed to properly choose the parameters α and β during the encoding process. Simulation results show that the proposed approach may offer better performance if compared with a fixed selection of parameters.

Acknowledgement

This research was supported by National Science Council of the Republic of China under Grant Number NSC-94-2219-E-009-008.

References

[1] *Streaming Video Profile-Final Draft Amendment (FDAM 4),* MPEG01/N3904.

[2] Weiping Li, "Overview of fine granularity scalability in mpeg-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 385-398, vol. 11, no. 3, March 2001.

[3] Hsiang-Chun Huang, Chung-Neng Wang, and Tihao Chiang, "A Robust Fine Granularity Scalability Using Trellis-Based Predictive Leak," IEEE Trans. On Circuits Syst. Video Technol., pp.372-385, vol. 12, No. 6, J

[4] L.C. Kuo, T. Chiang, S.J. Wang, "An Adaptive Macro-Block Based Scheme for Temporal Prediction Control in Fine Granularity Scalability Coding," in Conf, on WCE, pp. 7-13, Nov 2004, Hsinchu, Taiwan

[5] Mihaela van der Schaar, Hayder Radha, "A hybrid temporal-SNR Granular Scalability for Internet Video," IEEE Trans. On Circuits Syst. Video Technol., pp. 318-331, vol. 11, No. 3, March 2001

[6] Feng Wu, Shipeng Li, Ya-Win Zhang, "A framework for efficient progressive fine granularity scalable video coding," IEEE Trans. On Circuits Syst. Video Technol., pp.332-344, vol. 11, No. 3, March 2001

[7] Mihaela van der Schaar, Hayder Radha, "Motion-compensation fine-granular-scalability (MC-FGS) for wireless multimedia" Proceeding Multimedia Signal Processing, IEEE 4th Workshop pp. 453-458, on Oct. 2001

[8] S.R. Chen, C.P. Chang, and C.W. Lin, "MPEG-4 FGS coding performance improvement using inter-layer prediction," in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, pp. 265-268, May 2004, Montreal, Canada.

Adaptive QoS Control for Scalable Video Streaming over WLAN

Hee Young Cho, Doug Young Suh, and Gwang Hoon Park

{Media Lab.}, Kyunghee Univ., 1 Seocheon-dong Giheung-gu, Youngin-si, Kyunggido,
446-701, Korea
{marine0304, suh}@khu.ac.kr

Abstract. This paper is proposing a cross-layer optimization method in streaming video over WLAN. Packet losses of layers encoded by H.264 based scalable video codec are recovered by priority differentiated Reed Solomon erasure code. The optimal set of video layers to be transmitted and optimal FEC levels of video layers are determined under the constraints on channel condition time-varying with respect to packet loss ratio and available bandwidth.

Keywords: Cross-Layer, WLAN, Interference, Congestion, FEC.

1 Introduction

Wireless links which is changing rapidly can be characterized by Rayleigh fading, multi-path propagation by surrounding environments and interference.

It is an instance that existing TCP used in wired links performs poorly in a wireless environment because it isn't able to distinguish packet losses caused by network congestion or interference [1]. During last several years, most studies on cross-layer has been optimization only between physical layer and MAC layer or transport and MAC layer, which uses QoS parameters such as SNR (Signal to Noise Ratio), modulation, and so on [2]. Application layer has been included recently [3, 4].

This paper proposes to combine tools of different protocol layers to optimize use of limited resources, especially in wireless network. Zhao [5] and Angelis [6] introduced cross-layer optimization techniques in video streaming. Zhao optimized by using Lagrange optimization technique while Angelis used genetic algorithm technique. In this paper, optimization technique is applied to H.264/AVC based scalable layer video. In order to provide more realistic network condition, network parameters are derived from experiments on current commercial WLAN in harsh condition including interference of microwave oven and congestion by intentional background traffic.

Section 2 describes background knowledge needed to understand this paper. It is followed by proposed cross-layer optimization technique in Section 3. It, also, includes simulation result of each step. Effect of the proposed technique is discussed in Section 4 and Section 5 concludes this paper.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 949 – 958, 2006.

2 Video Services over WLAN

All services are assumed to share a channel (air) and to be controlled by a home server (an AP (access point) with multimedia server). The channel could be influenced by radiofrequency noise generated by microwave oven or neighboring APs.

2.1 H.264/SVC Based Scalable Video

H.264 is the most efficient video codec and used as the backbone technique in currently standardizing SVC (Scalable Video Coding) of JVT (Joint Video Team) by MPEG and ITU-T VCEG (Video Coding Experts Group). SVC [7] includes spatial, quality, and temporal scalable coding. By using the codec, a video sequence is encoded in bit-streams of 4 layers including B (base layer), En1 (enhancement 1 layer), En2, and En3, in which priority of B is the highest and that of En3 is the lowest. As shown in Table 1, "Soccer" sequence is encoded with hybrid scalability.

Table 1. Frame rate, Bit-rate r_l (kbps), packet size (bytes) n_l of every layer (Number of packets, k per second is assumed to be 70.)

	Base	En1	En2	En3
Frame rate [Hz]	15	15	30	30
Sequence Size	QCIF	QCIF	CIF	CIF
Bit-rate [kbps]	92.9	189.3	374.8	751.8
Packet Size [bytes]	170	346	685	1375

2.2 Network Condition and QoS (Quality of Service) Control

There are two sources of frame loss in WLAN, interference and congestion. Interference could be caused by micro-oven or other APs. It is assumed that available bandwidth remains constant and FLR (frame loss ratio) increases during interference. Congestion occurs due to increase in total traffic by other users or other services. During congestion, available bandwidth decreases and frames are lost because of buffer overflow, but not because of channel error. Accordingly, channel condition is classified into four states as following: good state (GS) and three bad states due to congestion (CBS), interference (IBS), and both (BBS) as shown in Table 2.

Table 2. Channel condition of 4 states

	GS	CBS	IBS	BBS
Available Rate (Mbps)	1.0	0.5	1.0	0.5
FLR	5%	5%	40%	40%

If the AP and the network card in terminal receive channel and network condition, their information about MAC layer can be used to tell why frames are lost and to take action to reduce frame loss ratio. Against interference, FER (Forward Erasure Recovery) technique can be used while TCP friendly approach can be used against congestion. If all services are capable of TCP friendly approach, traffic can be reduced as soon as congestion is detected.

If a frame loss is detected in transport layer, it is hard to tell what type of frame loss has been occurred. Then, the same action will be taken for either types of frame loss, interference or congestion. End-to-end delay results in two drawbacks after transition between good state and bad state. Transition from Bad to Good results in 'false alarm' period which happens when the available bandwidth is not fully used. On the other hand, transition from Good to Bad results in 'overflow' period which happens when the server sends data more than the available bandwidth. Length of 'false alarm' and 'overflow' periods can be reduced by using MAC layer parameters which is detected and informed in every 50ms.

3 Cross-Layer Optimization

In this paper, video service is optimized with respect to video quality and available bit-rate (R). An optimization is performed in various situations to select minimum distortion Đ(R, Δ). It is the expected image distortion at a certain set of R and Δ. The number of parity frames allocated to each transmitting layer is denoted by Δ. It is detailed in 3.4. All the possible sets of transmitting layers within R are tested and the set that has the lowest expected image distortion is selected.

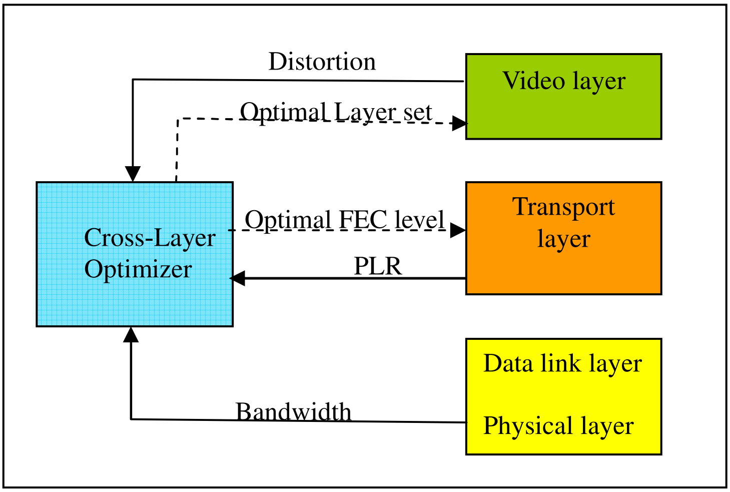


Fig. 1. Cross-Layer Optimization

3.1 Expected Video Quality Given (R, Δ)

We assume Base, Enhance 1, Enhance 2, and Enhance 3 layer as B, 1, 2, and 3, respectively. For example, if set s= {B, 1, 2, 3} is sent, the possible outcome sets that can be received at the receiver side are

$SS = \{\phi, B, B1, B2, B12, B3, B13, B23, B123, 1, 2, 3, 12, 13, 23\}$ .i.e., SS is the set, which has all subsets of elements.

$$D_s(R, \Delta) = \sum_{ss \in SS} \mathrm{P}_{ss} \mathrm{D}_{ss} \tag{1}$$

Where D_{ss}, P_{ss} are distortion value and loss probability of subset ss , respectively.

Bit stream is not decoded if inter-video layers are lost. For instance, it is supposed that all 4 layers are transmitted to the end device, if packets of all layers except En1 layer are received, the existing SVC decoder decodes only the Base layer, not En2 layer or En3 layer. However, as shown Figure 2, D_{B2} or D_{B23} have much better quality than D_B . i.e., we think that En2 and En3 are decoded with Base layer instead of throwing them away and decoding only Base layer. In case that most important Base layer is loss, error concealment in which a previous frame is duplicated.

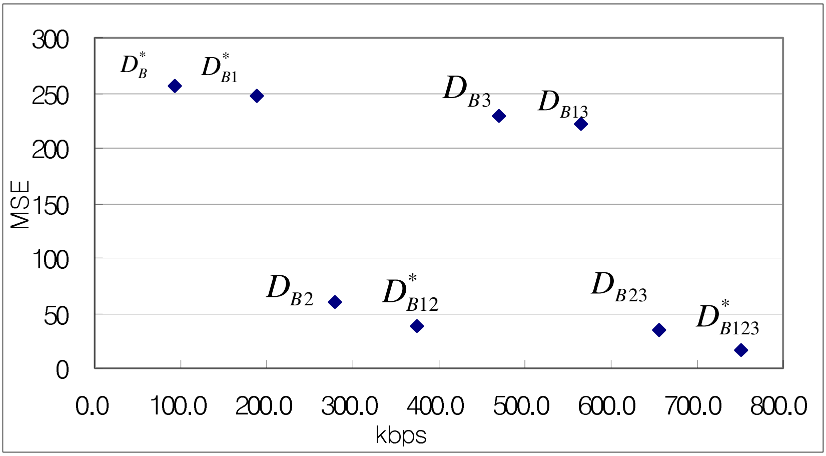


Fig. 2. Rate-Distortion of all possible sets of layers, one of sets with * is selected to be transmitted

If s is $\{B,1,2,3\}$, expected distortion is as below

$$\begin{aligned} D_{\{B,1,2,3\}}(R,\Delta) = & \sum_{ss \in \{\phi, B, 1, B1, \cdots B123\}} \mathrm{P}_{ss} \mathrm{D}_{ss} \\ & = P_\phi D_\phi + P_B D_B + P_1 D_1 + P_2 D_2 + P_3 D_3 + \\ & P_{12} D_{12} + P_{13} D_{13} + P_{23} D_{23} + P_{123} D_{123} + \\ & P_{B1} D_{B1} + P_{B2} D_{B2} + P_{B3} D_{B3} + P_{B12} D_{B12} + \\ & P_{B13} D_{B13} + P_{B23} D_{B23} + P_{B123} D_{B123} \end{aligned} \tag{2}$$

P_{ss} of subset $ss = \{B,1,2,3\}$ is as below equation (3). Where pb_l is residual loss probability of $l-th$ layer, and $(1-pb_l)$ is successful transmission probability of $l-th$ layer.

$$\begin{aligned}
&P_{\phi} = pb_B \cdot pb_1 \cdot pb_2 \cdot pb_3 \\
&P_1 = pb_B \cdot (1-pb_1) \cdot pb_2 \cdot pb_3 \\
&P_2 = pb_B \cdot pb_1 \cdot (1-pb_2) \cdot pb_3 \\
&P_{12} = pb_B \cdot (1-pb_1) \cdot (1-pb_2) \cdot pb_3 \\
&P_B = (1-pb_B) \cdot pb_1 \cdot pb_2 \cdot pb_3 \\
&P_{B1} = (1-pb_B) \cdot (1-pb_1) \cdot pb_2 \cdot pb_3 \\
&P_{B2} = (1-pb_B) \cdot pb_1 \cdot (1-pb_2) \cdot pb_3 \\
&P_{B12} = (1-pb_B) \cdot (1-pb_1) \cdot (1-pb_2) \cdot pb_3 \\
&\qquad\qquad \vdots \\
&P_{B123} = (1-pb_B) \cdot (1-pb_1) \cdot (1-pb_2) \cdot (1-pb_3)
\end{aligned} \tag{3}$$

Expected residual loss probability of $l-th$ layer pb_l can be calculated as shown equation (4). pb_l is mean value. $\binom{n}{i} q_l{}^i (1-q_l)^{n-i}$ is loss probability when i packets are lost.

$$pb_l = \sum_{i=n-k+1}^{n} \binom{n}{i} q_l{}^i (1-q_l)^{n-i} \frac{i}{n} \tag{4}$$

Where q_l is FLR (Frame Loss Probability) of $l-th$ layer; n is the number of total frame; k denotes number of data frame; $n-k$ is number of parity. In our simulation, k of every layer is the same.

3.2 Channel Condition

Channel condition is described by available bit-rate R and frame loss ratio P. Frame loss ratio is the same over all layers, but residual frame loss ratio of each layer is different since the parity level of each layer is different. In transport layer based control, only frame loss ratio is used as a channel condition parameter.

Channel condition is classified into four states, and they are Good State (GS), Interference-caused Bad State (IBS), Congestion-caused Bad State (CBS), and Bad State of both (BBS). Available bit-rate and frame loss ratio through air channel is given in Table 2.

If a service uses more than R, frames are lost at the same rate of excess bit-rate. Transition between four states can be modeled by 2 independent Gilbert Model which is a two-state Markov chain as in Figure 3. Transition probability can be determined in various environments. For example, P_{CG} is transition probability from CBS to GS. R_{GS} and P_{GS} are available bit-rate and FLR during GS, respectively.

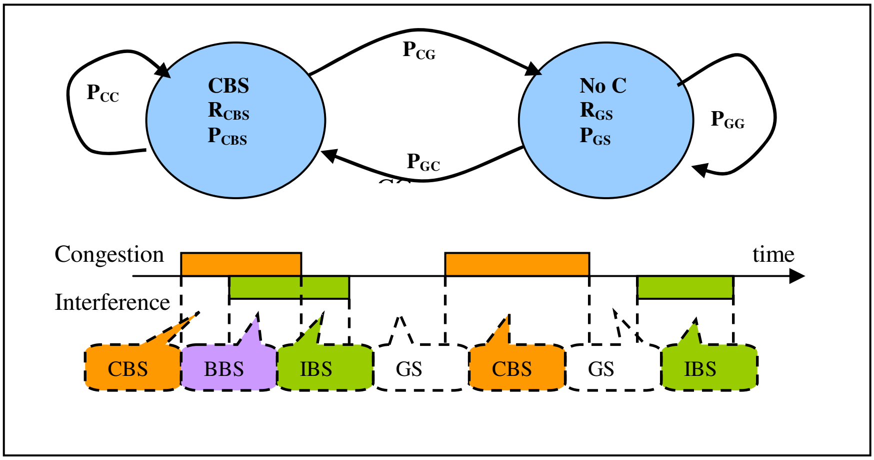


Fig. 3. Two states Markov Chain Model

3.3 Optimization During All States

Procedure to calculate the optimal variable set for each state is as follows,

1. Select a set of layers s, whose total bit-rate r is less than equal to R. (A set out of {B, B1, B2, B12, B123}) Then, R-r is left for parity.

2. Select the optimal Δ which minimizes distortion for the selected set s. Let Đ$_s$ and Δ_s* denote the minimum distortion for set s:

For each Δ, according to parity allocation method as in Figure 4, determine coding ratio for every layer, and then, calculate residual FLR of each layer and expected distortion.

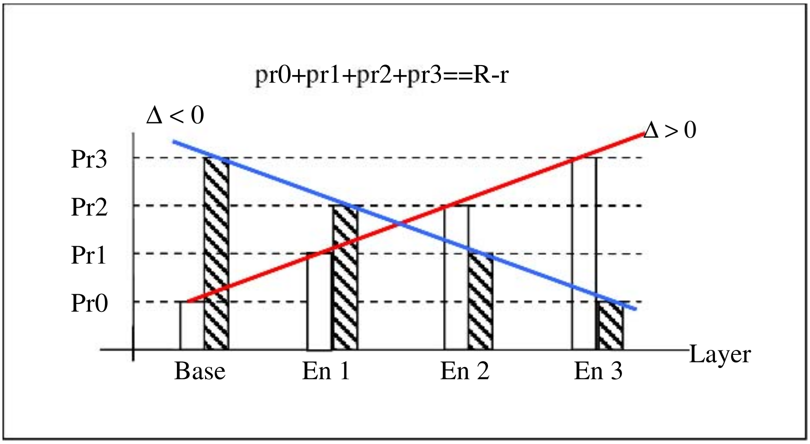


Fig. 4. parity allocation method

$$R - r = \sum_{i=0}^{|s|} (y_0 + \Delta i) \tag{5}$$

Where |s| is the number of layers in s and unique y_0 can be determined for given Δ.

3. Select the minimum Đ$_s$ among {Đ$_s$: s in {B, B1, B12, B123}}
For example, in case of IBS, since available bit-rates is 0.5Mbps, transmitting set of layers is selected among B, B1, and B12.

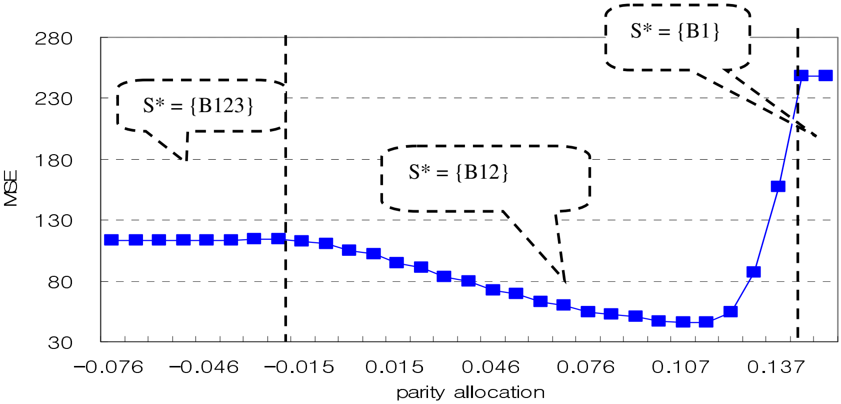


Fig. 5. Optimal parity allocation (Δ) is 0.107 and Optimal layer-set (s*) is {B12}, which is the minimum average MSE value, 46, at available bit-rate R=1.0Mbps and PLR = 0.4(IBS)

Figure 5 shows expected distortion vs Δ to find Đ$_s$, for s= {B12}. As shown in the figure, in case of IBS, layer set {B12} is transmitted instead of layer set {B123} since it improves video quality for sending parity rather than for higher layer data.

Table 3. Optimal transmitting set s*, expected distortion Đ$_s$ in PSNR (mse), and residual FLRs

	GS	CBS	IBS	BBS
s*	B123	B12	B12	B12
Đ$_s$	16	39	46	129
	0.03	0.023	1.107	-0.069
P_B	8.4e-7	2.2e-7	1.9e-4	2.0e-5
P_1	3.0e-9	2.2e-7	1.9e-4	4.0e-1
P_2	8.4e-7	3.5e-5	3.1e-2	4.0e-1
P_3	3.0e-4	NA	NA	NA

The optimal sets of layers are determined as shown in Table 3. In case of BBS, the PLR is higher than PLR of CBS. Therefore, as shown table 3, allocated parity amount of Base layer is larger than other Enhance layers.

3.4 Feedback Delay and QoS Control

It is assumed that the server receives RTCP RR (Receiver Report) every second and information in the RR packet is moving at the speed of the average of previous 3 seconds. It means that it takes more than 3 seconds for the server to get current channel condition. This gap results in two drawbacks after transition between good state and bad state. Transition from Bad to Good results in 'false alarm' period during which available bandwidth is not fully used. Transition from Good to Bad results in 'overflow' period during which the server sends data more than the available bandwidth. Length of 'false alarm' and 'overflow' periods can be reduced by using MAC layer parameters which is detected and informed in every 50ms.

4 Feedback Protocols

Feedback signals from MAC layer is much faster and more accurate. These two benefits are analyzed in this section. The benefits are proportional to state transition rate.

4.1 Effect of Faster Feedback

Table 4 and Figure 6 show the merits of using MAC layer feedback against transport layer feedback (e.g. RTCP). At transition from bad state to good state, faster feedback reduces duration of 'false alarm' period while at transition from good state to bad state it reduces duration of 'overflow' period.

Table 4. PSNR loss per picture due to late feedback

. Overflow		False alarm	
GS→ IBS	13.2 dB	IBS→GS	3.9dB
GS→ CBS	13.9dB	CBS→GS	3.9 dB
IBS→BBS	7.1 dB	BBS→IBS	4.5 dB
CBS→BBS	8.7 dB	BBS→CBS	1.1 dB

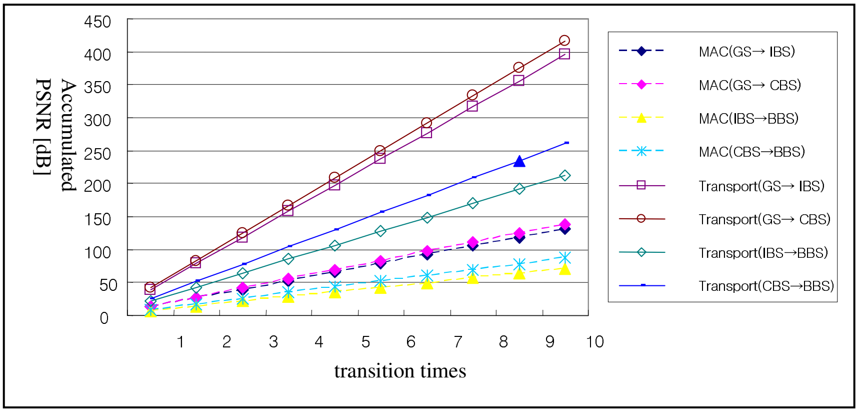


Fig. 6. Gain of faster feedback is proportional to feedback delay and number of transitions during a certain period

4.2 Effect of Accurate Signaling

It is clear that appropriate action against CBS is totally different from one against IBS. Amount of transmitted data packets must be reduced by dropping higher layers when congestion is occurred. The appropriate action against interference is adding more parity packets for frame recovery without reducing bit-rate. MAC layer can discern between CBS and IBS while transport layer can not, so we are studying about MAC layer parameters which can distinguish CBS and IBS, and then we will apply it to real streaming application.

5 Conclusion

Cross layer optimization technique for streaming video over WLAN. H.264 based scalable video coding is combined with differentiated frame recovery level with respect to priority of each video layer in transport layer and MAC layer feedback. For 4 pre-defined states of channel condition including ‘Good State,’ ‘Interference caused Bad State’, ‘Congestion caused Bad State,’ and ‘Bad State due to both,’ global optimal solutions are searched by using the steepest descent algorithm.

This paper also showed that MAC layer feedback outperforms transport layer feedback in two reasons. First, since it is faster, it helps the server to adapt faster to transition of channel condition. Second, since it can discern between frame loss caused by interference and congestion, the server can take a policy appropriate to the channel condition. This paper lacks in optimization of algorithm so that further research is required for real-time implementation.

Currently MAC layer is evolving to support various QoS functions such as priority control, bandwidth allocation, traffic dependent scheduling, repetition for lost frame, and etc. As incorporating such protocols with scalable video coding, cross layer optimization techniques will provide quantitative optimum under time varying network condition.

Acknowledgements. This work is supported by Korea Science & Engineering Foundation through the National Research Laboratory (NRL) program (Grant 2005-01363).

References

1. Ye Tian, Kai xu, Nirwan Ansari, “TCP in Wireless Environments: Problems and Solutions”, IEEE Radio Communications, S27-S32, March 2005.
2. Ivaylo Haratcherev, Jacco Taal, Koen Langerdoen, Reginald Lagendijk, Henk Sips, “Optimized Video Streaming over 802.11 by Cross-Layer Signaling.”, IEEE Communications Magazine, pp 115-122, 2006 Jan.
3. S. Khan, Y. Peng, E. Steinbach, M. Sgroi, W. Kellerer, “Application-driven cross-layer optimization for video streaming over wireless networks”, IEEE Communications Magazine, Volume 4, Issue: 1, pp122-130, 2006, Jan.

4. A. Ksentini, A. Gueroui, and M. Naimi, “Toward an Improvement of H.264Video Transmission over IEEE 802.11e through a Cross-Layer Architecture”, IEEE Communications Magazine, pp 107-114, 2006 Jan.
5. S. Zhao, Z. Xiong, X. Wang, "Joint error control and power allocation for video transmission over CDMA networks with multiuser detection," IEEE Tr. on CSVT, vol. 12, no. 6, June 2002.
6. F. Angelis, I. Habib, F. Davide, M. Naghshineh, "Intelligent content aware services in 3G wireless networks," IEEE JSAC, Vol. 23, No. 2, pp221-234, Feb. 2005.
7. JVT of ISO/IEC MPEG & ITU-T VCEG, N7555, Working Draft 4 of ISO/IEC 14496-10/AMD3 Scalable Video Coding, Oct. 2005.
8. J. Jeong, J. Shin, D. Y. Suh, "Quality enhancement of video services over QoS controlled networks," IEICE Tr. Comm., vol. E86-B, No. 2, pp562-571, Feb. 2003..
9. J. McDougall, S. Miller, "Sensitivity of wireless network simulation to a two-state Markov model channel approximation," GLOBECOM2003, pp697-701, 2003.

H.264 Intra Mode Decision for Reducing Complexity Using Directional Masks and Neighboring Modes

Jongho Kim, Kicheol Jeon, and Jechang Jeong

Dept. Electrical and Computer Engineering, Hanyang University,
17 Haengdang, Seongdong, Seoul 133-791, Korea
{angel, hobohemia, jjeong}@ece.hanyang.ac.kr

Abstract. The computational complexity for the RDO-based intra prediction in H.264 is very high due to many coding modes. There are 9 modes for 4×4 luma blocks, 4 modes for 16×16 luma blocks and 4 modes for 8×8 chroma blocks, respectively, so that all possible combinations of coding modes per MB are 592. We propose an intra mode decision algorithm for reducing complexity using directional masks and neighboring blocks' modes. The proposed algorithm has a very simple structure and reduces the number of mode combinations into 132 at the most. Simulation results show that the proposed algorithm reduces the encoding time up to 70% with negligible PSNR loss and bit-rate increase compared with exhaustive testing of H.264.

Keywords: intra mode selection, intra prediction, RDO, H.264/AVC.

1 Introduction

The emerging H.264/AVC is incorporated into new applications such as DMB (Digital Multimedia Broadcasting) and DVB-H on account of its outstanding coding performance which is known to be superior to MPEG-4 ASP (Advanced Simple Profile) by about 40% to 50% [1]. The H.264/AVC standard adopts a lot of state-of-the-art techniques to improve coding performance [2]: 4×4 block-based integer transform, motion compensation using variable block sizes and multiple references, advanced in-loop deblocking filter, improved entropy coders such as CAVLC (Context Adaptive VLC) and CABAC (Context Adaptive Binary Arithmetic Coding), and enhanced intra-prediction, etc. The RDO (Rate-Distortion Optimization) procedure is conducted in the intra- and inter-prediction of H.264/AVC to select the best coding mode among possible combinations. The best coding mode based on RDO means that the mode selected among possible combinations guarantees the smallest distortion under the given bit-rate instead of just minimizing the bit-rate or the distortion. The H.264/AVC encoder checks all possible combinations exhaustively to select the best coding mode based on RDO. Since RDO procedure should perform the transform and entropy coding for each coding mode, computational complexity is increased extremely compared to the conventional, thereby it makes H.264/AVC difficult to apply directly to low complexity devices. Many algorithms have been proposed to reduce the computational complexity, such as fast motion estimation [3,

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 959 – 968, 2006.

4] and fast inter mode selection algorithm [5, 6], etc. Fast motion estimation is a steady-studied subject through various standards and applications. On the other hand, fast mode selection for intra- and inter-prediction in H.264/AVC is a challenging subject. Since there are a lot of mode combinations for each macroblock, fast mode selection of intra- and inter-prediction plays an important role in reducing overall complexity. Intra coding is also carried out in inter-coded frames as well as intra-coded frames, thus fast intra mode selection is valuable for improving the overall coding performance of H.264/AVC.

We reduce complexity for the H.264/AVC intra-prediction using directional masks, which are used for detecting the directional correlation within a block, and neighboring modes. The proposed directional masks are used for 4×4 luma blocks, since there are many coding modes, on the other hand, we use the neighboring mode information for 16×16 luma blocks and 8×8 chroma blocks, since those blocks include a few number of coding modes and the blocks are relatively homogeneous. The directional masks are designed to represent each direction defined in H.264/AVC. We also address a sampling method in order to reduce computations of SATD (Sum of Absolute Transformed Difference) for 16×16 luma blocks.

The remaining parts of the paper are as follows. We review the intra-prediction scheme of H.264/AVC for 4×4 luma blocks, 16×16 luma blocks, and 8×8 chroma blocks, respectively, and RDO-based mode selection in Section 2. Section 3 presents, in detail, the proposed complexity reduction algorithm in intra mode selection, based on directional masks and neighboring mode information. Simulation results and conclusions are given in Section 4 and Section 5, respectively.

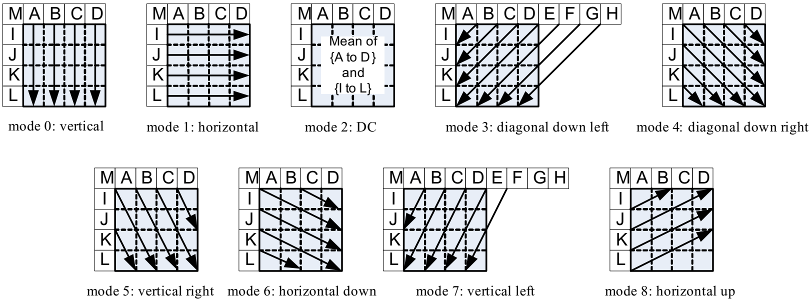
2 Intra Mode Selection in H.264/AVC

The H.264/AVC intra-prediction exploits the directional correlation with adjacent blocks in spatial domain, and selects the best mode by RDO among a lot of mode combinations. In this section, we review the intra mode selection for each block type (4×4 luma block, 16×16 luma block, and 8×8 chroma block) of H.264/AVC and address its computational complexity when the RDO procedure is used for the mode decision.

2.1 Intra-prediction in H.264/AVC

H.264/AVC adopts the directional intra-prediction in spatial domain with following advantages: the pixels in spatial domain are more correlated each other than the coefficients in transform domain, and the directional prediction can reflect local properties of images better. For the intra-prediction, we use boundary pixels of previously reconstructed neighboring blocks, which are upper, upper-right, and left blocks, and the current block is predicted to the maximum correlated direction. The H.264/AVC intra-prediction is conducted for all types of blocks: 4×4 luma blocks, 16×16 luma blocks, and 8×8 chroma blocks. For 4×4 luma blocks, which are selected mainly in non-homogeneous areas, there are 9 prediction modes, whereas for 16×16 luma blocks, which are selected in relatively homogeneous areas, there are

4 prediction modes. In addition, for 8×8 chroma blocks, there are 4 prediction modes, and the same mode is applied to two chrominance components (U and V). Note that two types of blocks, 16×16 luma blocks and 8×8 chroma blocks, have the same modes but the order of modes are different from each other.



mode 0: vertical
mode 1: horizontal
mode 2: DC
mode 3: diagonal down left
mode 4: diagonal down right
mode 5: vertical right
mode 6: horizontal down
mode 7: vertical left
mode 8: horizontal up

Fig. 1. 9 intra-prediction modes for a 4×4 luma block defined in H.264/AVC

Fig. 1 shows 9 intra modes for a 4×4 luma block, where *A* to *M* represents the boundary pixels of previously reconstructed adjacent blocks available at the time of prediction, and the arrows indicate the direction of prediction in each coding mode. DC prediction (*mode 2*), which is not directional mode, is performed using an average of *A* to *L*. For *mode 3* to *mode 8*, the pixels of current block are predicted using a weighted average of *A* to *M* along with the corresponding direction. Since 16×16 luma blocks are selected in relatively homogeneous areas mostly, there are fewer prediction modes, i.e., 4 modes of vertical (*mode 0*), horizontal (*mode 1*), DC (*mode 2*), and plane (*mode 3*) prediction. For 8×8 chroma blocks, there are also 4 prediction modes, which are same with the case of 16×16 luma prediction except the order of modes, such as DC (*mode 0*), horizontal (*mode 1*), vertical (*mode 2*), and plane (*mode 3*) prediction. To obtain the best mode among these modes, the H.264/AVC encoder carries out the rate-distortion optimization (RDO) procedure for each macroblock.

2.2 Selection of the Best Mode Using Rate-Distortion Optimization (RDO)

The RDO procedure for one macroblock in the intra-prediction is as follows [7, 8].

Initialization. Set parameters: macroblock quantization parameter QP and Lagrangian multiplier $\lambda_{MODE} = 0.85 \cdot 2^{(QP-12)/3}$ [9].

Step 1. For a 4×4 luma block, select the best mode, which minimizes the *Cost* of (1), among 9 modes.

$$Cost = \mathrm{D} + \lambda_{MODE} \cdot \mathrm{R}, \qquad (1)$$

where D and R denote distortion and bit-rate with given QP, respectively. MODE indicates one of the 9 intra modes of a 4×4 luma block. The distortion is obtained by SSD (Sum of Squared Difference) between the original 4×4 luma block and its reconstructed block, and the bit-rate includes the bits for the mode information and the transformed coefficients for the 4×4 luma block. Repeat this procedure for 16 4×4 luma blocks of a macroblock.

Step 2. For a 16×16 luma block, choose the mode that has the minimum SATD (Sum of Absolute Transformed Difference) among 4 modes as the best mode. In this case, we use Hadamard transform for SATD.

Step 3. For an 8×8 chroma block, select the best mode, which minimizes the $Cost_c$ of (2) among 4 modes.

$$Cost_c = \mathrm{D} + \lambda_{MODE} \cdot \mathrm{R}, \tag{2}$$

where D is obtained by SSD between two original 8×8 chroma blocks (U and V) and their reconstructed blocks. R, in this case, includes only the bits for the transformed coefficients unlike the 4×4 luma prediction case.

Step 4. Choose the best one as the prediction mode of one macroblock by comparing RD costs for 4×4 mode obtained from Step 1 and 16×16 mode from Step 2.

Considering the RDO procedure for intra mode selection in H.264/AVC, the number of mode combinations in one macroblock is $N_8 \times (16 \times N_4 + N_{16})$, where N_8, N_4, and N_{16} represent the number of modes of an 8×8 chroma block, a 4×4 luma block, and a 16×16 luma block, respectively. In other words, the H.264/AVC encoder carries out 592 RDO calculations to select the best mode for one. As a result, the complexity of the encoder increases extremely. We propose, in next section, a complexity reduction algorithm in the H.264/AVC intra-prediction by reducing the number of RDO calculations for intra mode selection without visual quality degradation.

3 Proposed Intra Mode Selection Algorithm

Since the RDO procedure includes time-consuming processes such as transform and entropy coding, the number of RDO computations is a critical point in the overall encoding speed. In this section, we describe a method to reduce the number of RDO computations for each block type.

3.1 Intra Mode Selection for 4×4 Luma Blocks

Two major observations on features of 4×4 luma block are found as follows: First, the directional correlation of a block is generally consistent with edge directions. Second, the prediction mode of current block is highly correlated with the modes of adjacent blocks.

From the first observation, we obtain one candidate mode using the proposed directional masks shown in Fig. 2, where black dots indicate pixel positions to be computed for directional correlation, and arrows represent the directions of correlation in each corresponding mask. Since the H.264/AVC intra-prediction defines 8 directions except DC mode, we propose 8 directional masks instead of precise edge detectors. We select one candidate mode with the minimum *Diff* using

$$Diff = |a - m| + |b - n| + |c - o| + |d - p|, \quad \text{for vertical direction,} \tag{3}$$

$$Diff = |a - d| + |e - h| + |i - l| + |m - p|, \quad \text{for horizontal direction,} \tag{4}$$

$$Diff = |c - i| + 2\cdot|d - m| + |h - n|, \qquad \text{for diagonal down left direction,} \tag{5}$$

$$Diff = |b - l| + 2\cdot|a - p| + |e - o|, \qquad \text{for diagonal down right direction,} \tag{6}$$

$$Diff = |a - n| + 2\cdot|b - o| + |c - p|, \qquad \text{for vertical right direction,} \tag{7}$$

$$Diff = |a - h| + 2\cdot|e - l| + |i - p|, \qquad \text{for horizontal down direction,} \tag{8}$$

$$Diff = |b - m| + 2\cdot|c - n| + |d - o|, \qquad \text{for vertical left direction,} \tag{9}$$

$$Diff = |e - d| + 2\cdot|i - h| + |m - l|, \qquad \text{for horizontal up direction,} \tag{10}$$

where *a* to *p* denotes the pixels represented to black dots in Fig. 2. Actual indices of pixel positions used in (3) to (10) are shown in Fig. 3(a). We regard the direction with smaller *Diff* as more correlated one.

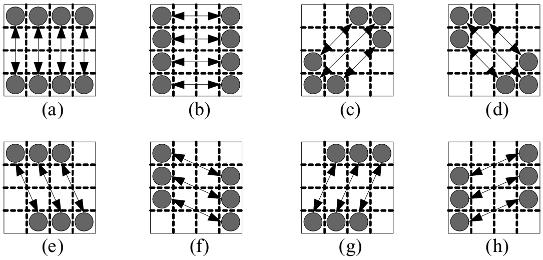
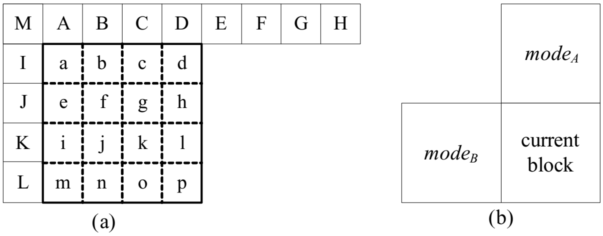


Fig. 2. The proposed directional masks for a 4×4 luma block. (a) vertical, (b) horizontal, (c) diagonal down left, (d) diagonal down right, (e) vertical right, (f) horizontal down, (g) vertical left, (h) horizontal up mask.



M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Fig. 3. Pixel indices and neighboring modes used in the proposed intra mode selection algorithm. (a) indices used in (3) to (10) for a 4×4 luma block, (b) modes of upper and left blocks for additional candidate modes.

From the second observation, we obtain additional candidate modes using neighboring mode information, where one is upper block's mode, $mode_A$, and the other is left block's mode, $mode_B$, as shown in Fig. 3(b). Since the coding modes of H.264/AVC intra-prediction are determined using adjacent blocks instead of just within the current block, we include the additional modes, $mode_A$ and $mode_B$, for candidate modes in RDO procedure. We include one additional mode when $mode_A$ and $mode_B$ are same, or two additional modes when $mode_A$ and $mode_B$ are different from each other, for RDO procedure.

To determine whether DC mode should be included in RDO procedure or not, we define S in (11) as a sum of difference between the average and each pixel of current block.

$$S = \sum_{i=0}^{15} \left| avg - p_i \right| \tag{11}$$

where $avg = \left(\sum_{i=0}^{15} p_i + 8 \right) \gg 4$ and p_i is each pixel of current block. If S is smaller than a threshold, T_1, we carry out RDO for at most 4 candidate modes, i.e., one mode from the proposed masks, at most two modes from adjacent blocks, and DC mode. If S is larger than a threshold, T_1, we carry out RDO for at most 4 candidate modes, i.e., two modes from the proposed masks (with minimum and second minimum *Diff*) and at most two modes from adjacent blocks. The proposed intra mode selection algorithm for a 4×4 luma block is summarized as follows.

Step 1. For a 4×4 luma block, obtain *avg* and *S* by (11).
Step 2a. If *S* is larger than a threshold, T_1, carry out RDO procedure for at most 4 candidate modes: two modes with minimum and second minimum *Diff* by (3) to (10), and at most two modes from adjacent blocks. In this case, DC mode of adjacent blocks is excluded from RDO procedure.
Step 2b. If *S* is smaller than a threshold, T_1, carry out RDO procedure for at most 4 candidate modes: one mode with minimum *Diff* by (3) to (10), at most two modes from adjacent blocks, and DC mode.

3.2 Intra Mode Selection for 16×16 Luma and 8×8 Chroma Blocks

The H.264/AVC intra-prediction selects 16×16 luma blocks when the area to be predicted is relatively homogeneous. The chrominance components of 4:2:0 format are also relatively homogeneous due to down-sampling and filtering. Thus, for 16×16 luma blocks and 8×8 chroma blocks, there are 4 coding modes, different from the case of 4×4 luma blocks. For intra mode selection with 16×16 luma blocks and 8×8 chroma blocks, we carry out RDO procedure using the neighboring modes not using directional masks. Since all adjacent blocks are not 16×16 in this case, we should consider some conditions such as sizes and modes of adjacent blocks when we select the best mode for 16×16 luma blocks. The proposed intra mode selection algorithm for a 16×16 luma block is summarized as follows.

Step 1. Examine the sizes of adjacent blocks: if both blocks (upper block and left block) are 16×16, go to Step 2, otherwise go to Step 4.
Step 2. Examine the modes of adjacent blocks: if both modes are same, go to Step 3, otherwise select the best mode for a 16×16 luma block, which results in the minimum SATD between two adjacent modes of $mode_A$ and $mode_B$.

Step 3. If both adjacent modes are DC mode, go to Step 4, otherwise select the best mode for a 16×16 luma block, which results in the minimum SATD between the neighboring mode and DC mode.

Step 4. Let Δ_V be a vertical difference between upper boundary pixels of the current block and boundary pixels of the upper block, and Δ_H be a horizontal difference between left boundary pixels of the current block and boundary pixels of the left block as follows.

$$\Delta_V = \sum_{i=0}^{15} |u_i - q_i|, \qquad \Delta_H = \sum_{i=0}^{15} |l_i - r_i|, \tag{12}$$

where u_i and q_i denote boundary pixels of the upper block and upper boundary pixels of the current block, respectively, and l_i and r_i denote boundary pixels of the left block and left boundary pixels of the current block, respectively, as depicted in Fig. 4(a). Obtain candidate modes using Δ_V and Δ_H: if $|\Delta_V - \Delta_H|$ is smaller than T_2, candidate modes are DC mode and plane mode; if $(\Delta_V - \Delta_H)$ is larger than T_2, candidate modes are DC mode and horizontal mode; if $(\Delta_V - \Delta_H)$ is smaller than $-T_2$, candidate modes are DC and vertical mode, where T_2 is a positive value. Finally, select the best mode between each candidate mode by choosing the mode with minimum SATD.

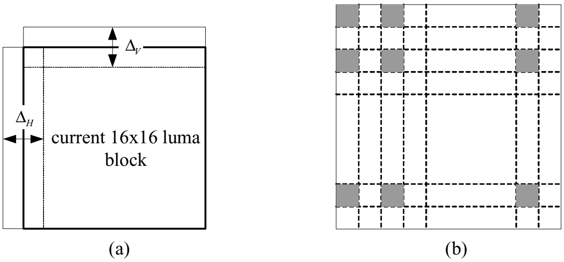


Fig. 4. For intra-prediction with a 16×16 luma block, (a) definition of Δ_V and Δ_H, (b) sampling method to reduce computations

To reduce SATD computation, we propose sampling method as illustrated in Fig. 4(b), where only shaded pixels are used for SATD. Since a 16×16 luma block is relatively homogeneous, the mode selection using sampled pixels has almost same results with the mode selection using all pixels. For 8×8 chroma blocks, a similar method to the method for 16×16 luma block is applied except examining whether the sizes of both adjacent blocks are same or not, since all adjacent blocks have the same size in this case.

Table 1. Comparison of the number of RDO computations

Block type	H.264/AVC method	Proposed method
4×4 luma block	9	at most 4
16×16 luma block	4	2
8×8 chroma block	4	2

Table 1 summarizes the number of candidate modes for RDO procedure in the proposed method. As it can be seen from Table 1, the proposed algorithm carries out only 132 RDO computations at the most, which are much less than those of exhaustive search in H.264/AVC video coding, i.e., 592 RDO computations.

4 Simulation Results

In order to evaluate the proposed algorithm, we used JM 8.4 (Joint Model ver. 8.4) under H.264/AVC baseline profile, which does not contain B-slice and CABAC, with RD optimization and Hadamard transform turned on. According to the test conditions specified in [10], we carried out simulations for test sequences of *Akiyo*, *Foreman*, *Carphone*, *Hall Monitor*, *Silent*, *News*, *Container*, and *Coastguard* with QCIF (176×144) resolution. We use various QP of 28, 32, and 40 with IPPP…type and I-only type, respectively. In IPPP…type, all 300 frames are tested for each sequence, where all frames are inter-coded with one intra-frame for every 100 frames; in I-only type, all 300 frames are intra-coded. The thresholds, which include T_1 to select the mode for a 4×4 luma block and T_2 to select the mode for a 16×16 luma block and an 8×8 chroma block, are set to 32 and 8, respectively. We compared the results with the case of exhaustive search in terms of the change of average PSNR (ΔPSNR), average data bits (ΔBit), and average encoding time (ΔTime), respectively, with the machine of Intel Pentium IV processor of 2.8 GHz and 512MB memory.

Table 3 summarizes the simulation results of the proposed algorithm for IPPP type of each sequence. In addition, Table 2 shows the results of F. Pan et al.'s method [6] as a reference for comparison. In Table 3, the minus of ΔPSNR and ΔTime means that the encoding time and PSNR are reduced compared with JM, respectively. It can be seen that the proposed algorithm saves the encoding time up to about 35% with negligible loss in PSNR and increment in bits. By comparing Tables 2 and 3, we can see that the proposed algorithm is superior to F. Pan et al.'s method. This is because F. Pan et al.'s method, to reduce the RDO computations, performs the quite complex pre-processing, and does not use the neighboring mode information.

Table 2. Results of F. Pan et al.'s method for comparison

Sequence (QCIF)	IPPP type			I-only type		
	ΔTime (%)	ΔPSNR (dB)	ΔBit (%)	ΔTime (%)	ΔPSNR (dB)	ΔBit (%)
Akiyo	-22.72	-0.053	1.17	-64.32	-0.210	3.21
Foreman	-21.80	-0.077	1.54	-65.38	-0.285	4.44
Carphone	-20.51	-0.082	1.80	-65.93	-0.276	3.91
Hall Monitor	-23.38	-0.065	1.23	-66.51	-0.252	3.73
Silent	-21.94	-0.033	0.86	-65.17	-0.183	3.54
News	-23.11	-0.067	1.23	-55.34	-0.294	3.90
Container	-20.78	-0.081	1.80	-56.36	-0.234	3.70
Coastguard	-21.20	-0.017	0.50	-55.03	-0.106	2.36

Table 3. Simulation results for IPPP type sequences

Sequence (QCIF)	QP = 28			QP = 32			QP = 40		
	ΔTime (%)	ΔPSNR (dB)	ΔBit (%)	ΔTime (%)	ΔPSNR (dB)	ΔBit (%)	ΔTime (%)	ΔPSNR (dB)	ΔBit (%)
Akiyo	-29.65	-0.013	0.16	-32.94	-0.008	0.21	-35.56	-0.002	0.31
Foreman	-30.61	-0.016	0.22	-34.37	-0.010	0.28	-36.17	-0.001	0.34
Carphone	-30.87	-0.018	0.17	-34.52	-0.013	0.32	-35.91	-0.009	0.33
Hall Monitor	-33.12	-0.018	0.20	-35.31	-0.014	0.26	-38.35	-0.005	0.37
Silent	-32.05	-0.014	0.18	-34.23	-0.008	0.25	-36.48	-0.004	0.30
News	-33.54	-0.017	0.19	-35.81	-0.011	0.25	-37.63	-0.003	0.30
Container	-29.93	-0.018	0.16	-33.42	-0.011	0.22	-35.41	-0.003	0.27
Coastguard	-30.05	-0.010	0.12	-33.62	-0.007	0.18	-35.42	-0.001	0.24

Table 4. Simulation results for I-only type sequences

Sequence (QCIF)	QP = 28			QP = 32			QP = 40		
	ΔTime (%)	ΔPSNR (dB)	ΔBit (%)	ΔTime (%)	ΔPSNR (dB)	ΔBit (%)	ΔTime (%)	ΔPSNR (dB)	ΔBit (%)
Akiyo	-62.34	-0.10	0.47	-67.16	-0.12	0.93	-68.85	-0.07	1.62
Foreman	-62.91	-0.08	0.15	-68.11	-0.07	1.06	-70.32	-0.03	1.79
Carphone	-65.28	-0.16	0.92	-67.84	-0.13	1.52	-69.13	-0.10	1.73
Hall Monitor	-66.12	-0.14	0.36	-69.63	-0.12	1.92	-71.42	-0.10	2.94
Silent	-63.35	-0.12	0.51	-67.35	-0.08	1.35	-60.04	-0.05	2.63
News	-61.38	-0.11	0.86	-66.56	-0.10	1.26	-69.28	-0.06	1.85
Container	-61.93	-0.09	0.90	-67.07	-0.08	1.07	-69.21	-0.05	1.71
Coastguard	-60.82	-0.08	0.73	-65.81	-0.06	0.82	-68.23	-0.03	1.57

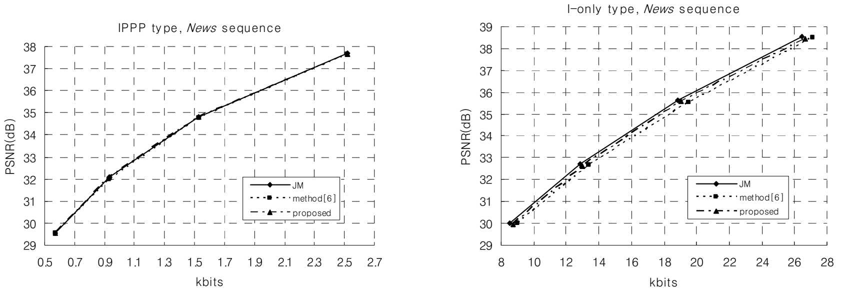


Fig. 5. R-D curve for *News* sequence, left: QCIF, IPPP type; right: QCIF, I-only type

We also show the simulation results for I-only type sequences in Table 4. It can be seen that the proposed algorithm saves the encoding time up to about 70%, since all frames are intra-coded. On the other hand, the drop in PSNR and the increase of bit-rate is somewhat larger than the IPPP case. However, these results can be acceptable because the drop in PSNR and the increase of bit-rate of I-only case are considerably small with respect to saving the encoding time, and I-only case is regarded as an extreme case. By comparing Tables 2 and 4, we can see that the proposed algorithm is superior to F. Pan et al.'s method for the same reasons of the IPPP case.

Fig. 5 shows the R-D curve, which includes the results of JM, F. Pan et al., and the proposed method, for IPPP type and I-only type of *News* sequence, respectively. We can see that the proposed algorithm is superior to JM and F. Pan et al.'s method as similar to Tables 3 and 4.

5 Conclusions

This paper has presented a complexity reduction algorithm for intra mode selection in H.264/AVC based on directional masks and neighboring mode information. The proposed directional masks have simple structures, which require no multiplications. The simulation results show that the proposed algorithm reduces the number of mode combinations and computational complexity for RDO with negligible loss of PSNR and bit-rate increment. The proposed algorithm can be applied to the H.264/AVC video encoder with low computational capability.

Acknowledgments. This work was supported by the research fund of Hanyang University (HY-2004-I).

References

1. Wiegand, T., Sullivan, G., Bjontegaar, G., Luthra, A.: Overview of the H.264/AVC Video Coding Standard. IEEE Trans. Circuits and Syst. for Video Technol., Vol. 13. (2003) 560-576
2. ITU-T Rec. H.264 | ISO/IEC 14496-10: Information Technology – Coding of Audio-visual Objects, Part 10: Advanced Video Coding (2002)
3. Chen, Z., Zhou, P., He, Y.: Fast Integer Pel and Fractional Pel Motion Estimation for JVT. Doc. JVT-F017 (2002)
4. Hsieh, B., Huang, Y., Wang, T., Chien, S., Chen, L.: Fast Motion Estimation for H.264/MPEG-4 AVC by Using Multiple Reference Frame Skipping Criteria. VCIP 2003, Proceedings of SPIE, Vol. 5150. (2003) 1551-1560
5. Lim, K., Wu, S., Wu, D., Rahardja, S., Lin, X., Pan, F., Li, Z.: Fast Inter Mode Decision. Doc. JVT-I020 (2003)
6. Pan, F., Lin, X., Rahardja, S., Lim, K., Li, Z., Wu, D., Wu, S.: Fast Mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding. IEEE Trans. Circuits and Syst. for Video Technol., Vol. 15. (2006) 813-822
7. Kim, C., Shih, H., Kuo, C.: Multistage Mode Decision for Intra Prediction in H.264 Codec. VCIP 2004, Proceedings of SPIE, Vol. 5308 (2004) 355-363
8. Lim, K., Sullivan, G., Wiegand, T.: Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods. Doc. JVT-N046 (2005)
9. Stockhammer, T., Kontopodis, D., Wiegand, T.: Rate-distortion Optimization for JVT/H.26L Video Coding in Packet Loss Environment. Int. Packet Video Workshop (2002)
10. Sullivan, G.: Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low Resolution Progressive Scan Source Material. Doc. VCEG-N81 (2001)

Seamless Channel Transition for BroadCatch Scheme

Yukon Chang and Shou-Li Hsu

Department of Information Engineering
I-Shou University, Kaohsiung County, Taiwan, 84001
yukon@isu.edu.tw, d9203002@stmail.isu.edu.tw

Abstract. BroadCatch belongs to a group of periodic broadcasting schemes that deliver popular video to a large number of users using multicasting. Some of the schemes have been shown to support seamless channel transition, a process that increases or decreases the number of channels allocated to a video without causing service interruption. By reallocating channels among movies depending on their popularity, the performance of a Video-on-Demand service can be improved. In this paper, we propose a seamless channel transition scheme based on BroadCatch and demonstrate its effectiveness as a performance tuning tool with simulation.

Keywords: broadcasting, channel allocation, BroadCatch, Video-on-Demand (VOD).

1 Introduction

Multicasting, a network technology which sends out the same packet to a group of clients with minimum duplication of packets at intermediate routers, provides a CPU- and bandwidth-efficient alternative to unicasting. Near Video-on-Demand (NVoD) adopts the multicast technique by partitioning video content into segments and broadcasting them repeatedly on multiple multicast channels in some particular arrangements. Examples of NVoD include batching broadcasting [1], fast broadcasting [2], pyramid broadcasting [3], skyscraper broadcasting [4], BroadCatch [5], staircase broadcasting [6], and harmonic broadcasting [7]. In any of these schemes, a user receives data from several channels simultaneously while playback of the movie is in progress. To ensure uninterrupted viewing, it is important for the client to have a segment ready, either directly from the network or from its buffer, at the moment when the segment is to be played back. Naturally, playback cannot start until the beginning of the first segment appears on its channel. This initial waiting time depends on the way segments are assigned to channels as well as the total number of channels allocated to this movie. More channels results in shorter waiting time and more viewer satisfaction.

A large NVoD system typically serves many movies. Since not all movies are equally popular, it is desirable to be able to reassign channels allocated to a less popular movie to a more popular one so that more viewers enjoy a shorter waiting time and the system has a lower overall average waiting time. Channel transition refers to the process of adjusting the number of channels allocated to a movie while it

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 969 – 978, 2006.

is being served. A channel transition scheme is said to be seamless if the adjustment of channels satisfies the following two conditions. First, any ongoing service should not be interrupted or broken off during the transition. Second, the maximum startup latency should be limited by an acceptable value during the transition. Channel transition for NVoD has been studied in [8-11].

In this paper, we show that seamless channel transition can be accomplished for BroadCatch scheme and study its effectiveness. The rest of this paper is organized as follows. In section 2, we give a short review of the original BroadCatch scheme as well as seamless channel transition for fast broadcasting. In section 3, the proposed seamless channel transition for BroadCatch scheme is presented and then compared with that for fast broadcasting. In section 4, we analyze the effectiveness of channel transition in a BroadCatch system with simulation. Conclusions are given in section 5.

2 Related Works

2.1 BroadCatch Scheme

Let N be the number of channels allocated to a video and name these channels $C_1, C_2, ..., C_N$. Equally divide the video into 2^{N-1} segments, $S_1, S_2, ..., S_{2^{N-1}}$. In other words, the entire video can be written as $S_1 \circ S_2 \circ ... \circ S_{2^{N-1}}$, where $\circ$ is the concatenation operation. Channel C_i repeatedly broadcasts a sequence of segments G_i at the playback rate, where

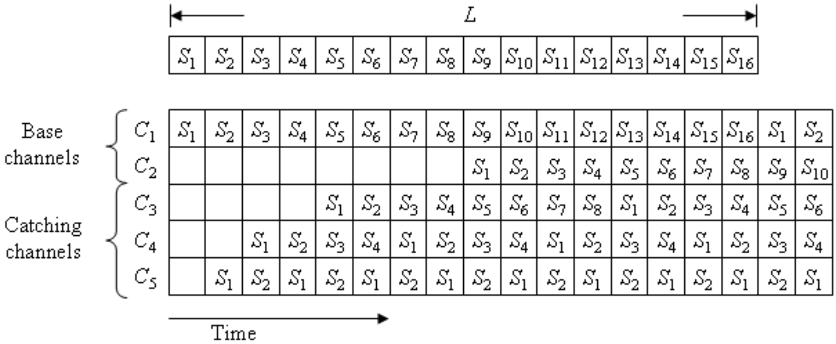
$$G_i = \begin{cases} S_1 \circ S_2 \circ ... \circ S_{2^{N-1}} & , \textit{if } i = 1,2 \\ S_1 \circ S_2 \circ ... \circ S_{2^{N-i+1}} & , \textit{if } 3 \le i \le N \end{cases} \tag{1}$$

Furthermore, the broadcast of G_i on channel C_i starts O_i segments after the broadcast on the channel C_1 has begun, where

$$O_i = \begin{cases} 0 & , \textit{if } i = 1 \\ 2^{N-2} & , \textit{if } i = 2 \\ \dfrac{O_{i-1}}{2} & , \textit{if } 2 < i \le N \end{cases} \tag{2}$$

Since the entire movie is partitioned into 2^{N-1} segments, the worst case waiting time is $L/2^{N-1}$ and average waiting time is $L/2^N$, where L is the length of the movie. One special property of BroadCatch is that although the video is broadcast on N channels, the client receives data from no more than $N-2$ channels at any given time. In other words, while the server side bandwidth requirement is N, the client side bandwidth requirement is only $N-2$.

Fig. 1 shows an example of the BroadCatch channel assignment when $N = 5$. Channels C_1 and C_2, called base channels, broadcast the entire video segments, while channels C_3, C_4, and C_5, called catching channels, broadcast G_3, G_4, and G_5 with offsets O_3, O_4, and O_5, respectively.



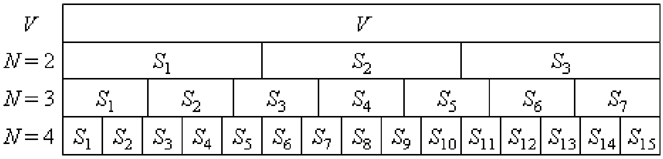
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}		
C_1	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_1	S_2
C_2									S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
C_3					S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_1	S_2	S_3	S_4	S_5	S_6
C_4			S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4
C_5		S_1	S_2	S_1	S_2	S_1	S_2	S_1	S_2	S_1	S_2	S_1	S_2	S_1	S_2	S_1	S_2	S_1

Fig. 1. BroadCatch channel assignment ($N = 5$)

2.2 Seamless Channel Transition on Fast Broadcasting

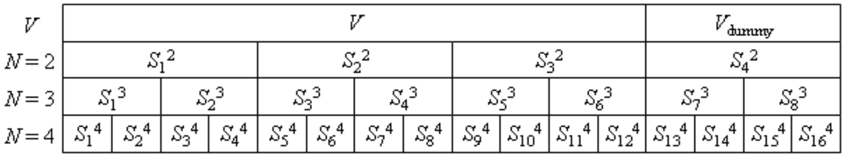
The basic idea of seamless channel transition scheme presented in [8] is to build a multiple relationship among segments with different channel assignments. Since the desired relationship is not present in the original fast broadcasting, shown in Fig. 2, a dummy video is added at the end of the movie. The augmented video can be partitioned into $3 \times 2^{N-2}$ segments for any N, $N \geq 2$, as shown in Fig. 3. Let C_i^N and S_j^N denote the i^{th} channel and j^{th} segment with N channels allocated to the video, respectively. With the new partitioning scheme, seamless channel transition now becomes possible because a segment at a smaller N is always equal to the concatenation of some segments at a larger N. For instance, $S_1^2 = S_1^3 \circ S_2^3 = S_1^4 \circ S_2^4 \circ S_3^4 \circ S_4^4$.

Fig. 4 shows the segment arrangements for $N = 2$, 3 and 4 in seamless channel transition. Channel transitions are categorized into positive and negative channel transitions, corresponding to adding and releasing channels, respectively. Tseng *et al.* went through elaborate arguments to show why viewers will not experience disruption at positive channel transition and what needs to be made up to viewer at negative channel transition.



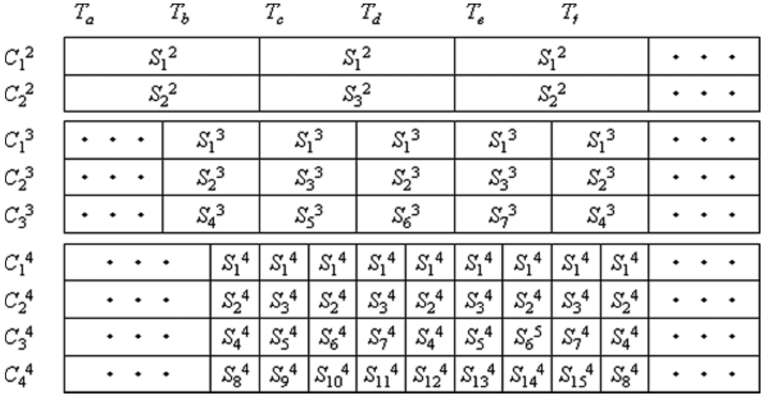
V
V
N=2
S_1 S_2 S_3
N=3
S_1 S_2 S_3 S_4 S_5 S_6 S_7
N=4
S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9 S_{10} S_{11} S_{12} S_{13} S_{14} S_{15}

Fig. 2. Original segmentations in fast broadcasting (N = 2, 3 and 4)



V	V												V_{dummy}			
$N=2$	S_1^2				S_2^2				S_3^2				S_4^2			
$N=3$	S_1^3		S_2^3		S_3^3		S_4^3		S_5^3		S_6^3		S_7^3		S_8^3	
$N=4$	S_1^4	S_2^4	S_3^4	S_4^4	S_5^4	S_6^4	S_7^4	S_8^4	S_9^4	S_{10}^4	S_{11}^4	S_{12}^4	S_{13}^4	S_{14}^4	S_{15}^4	S_{16}^4

Fig. 3. Segments after data padding with length $L/3$ when $\alpha = 2$



T_a
T_b
T_c
T_d
T_e
T_f

C_1^2	S_1^2	S_1^2	S_1^2	• • •
C_2^2	S_2^2	S_3^2	S_2^2	• • •

C_1^3	• • •	S_1^3	S_1^3	S_1^3	S_1^3	S_1^3	• • •
C_2^3	• • •	S_2^3	S_3^3	S_2^3	S_3^3	S_2^3	• • •
C_3^3	• • •	S_4^3	S_5^3	S_6^3	S_7^3	S_4^3	• • •

C_1^4	• • •	S_1^4	S_1^4	S_1^4	S_1^4	S_1^4	S_1^4	S_1^4	S_1^4	S_1^4	• • •
C_2^4	• • •	S_2^4	S_3^4	S_2^4	S_3^4	S_2^4	S_3^4	S_2^4	S_3^4	S_2^4	• • •
C_3^4	• • •	S_4^4	S_5^4	S_6^4	S_7^4	S_4^4	S_5^4	S_6^5	S_7^4	S_4^4	• • •
C_4^4	• • •	S_8^4	S_9^4	S_{10}^4	S_{11}^4	S_{12}^4	S_{13}^4	S_{14}^4	S_{15}^4	S_8^4	• • •

Fig. 4. Segment arrangements for $N = 2$, 3 and 4 in seamless channel transition

Note that the waiting time in this scheme is longer than that in the original fast broadcast because of the insertion of the dummy video. In addition, a segment may be moved from one channel to a different channel after a channel transition, making the channel transition logic rather complicated.

3 Seamless Channel Transition Scheme for BroadCatch

In this section, we first present a seamless channel transition scheme for BroadCatch and then compare our scheme with the one described in section 2.2. To make the description and comparison easy, we adopt the notation used in [8].

3.1 Seamless Channel Transition

Let S_j^N denote the j^{th} segment of the video when using N channels. Using this notation, the original BroadCatch channel segmentation using $N = 2$, 3, 4, and 5 channels is shown in Fig. 5. Note that the desired relationship among segments, namely, $S_j^N = S_{2j-1}^{N+1} \circ S_{2j}^{N+1}$, is still present. Also, let G_i^N denote the sequence of segments delivered on channel i using N channels, that is,

$$G_i^N = \begin{cases} S_1^N \circ S_2^N \circ \ldots \circ S_{2^{N-1}}^N & , \textit{ if } i = 1,2 \\ S_1^N \circ S_2^N \circ \ldots \circ S_{2^{N-i+1}}^N & , \textit{ if } 3 \le i \le N \end{cases} \tag{3}$$

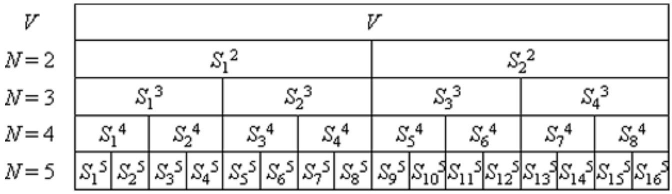
Note that since $S_j^N = S_{2j-1}^{N+1} \circ S_{2j}^{N+1}$, G_i^N is identical to G_i^{N+1}. Furthermore,

$$G_i^N = S_1^N \circ S_2^N \circ \ldots \circ S_{k-1}^N \circ S_k^N \circ S_{2k+1}^{N+1} \circ S_{2k+2}^{N+1} \circ \ldots \circ S_{2^{N-i+2}-1}^{N+1} \circ S_{2^{N-i+2}}^{N+1} = G_i^{N+1} \tag{4}$$

and

$$G_i^{N+1} = S_1^{N+1} \circ S_2^{N+1} \circ \ldots \circ S_{2k-1}^{N+1} \circ S_{2k}^{N+1} \circ S_{k+1}^N \circ S_{k+2}^N \circ \ldots \circ S_{2^{N-i+1}-1}^N \circ S_{2^{N-i+1}}^N = G_i^N \tag{5}$$

for all i, k, and N, where $1 \le i \le N$, $1 \le k \le 2^{N-i+1}$.



<table>
<tr><td>V</td><td colspan="16">V</td></tr>
<tr><td>N=2</td><td colspan="8">S_1^2</td><td colspan="8">S_2^2</td></tr>
<tr><td>N=3</td><td colspan="4">S_1^3</td><td colspan="4">S_2^3</td><td colspan="4">S_3^3</td><td colspan="4">S_4^3</td></tr>
<tr><td>N=4</td><td colspan="2">S_1^4</td><td colspan="2">S_2^4</td><td colspan="2">S_3^4</td><td colspan="2">S_4^4</td><td colspan="2">S_5^4</td><td colspan="2">S_6^4</td><td colspan="2">S_7^4</td><td colspan="2">S_8^4</td></tr>
<tr><td>N=5</td><td>S_1^5</td><td>S_2^5</td><td>S_3^5</td><td>S_4^5</td><td>S_5^5</td><td>S_6^5</td><td>S_7^5</td><td>S_8^5</td><td>S_9^5</td><td>S_{10}^5</td><td>S_{11}^5</td><td>S_{12}^5</td><td>S_{13}^5</td><td>S_{14}^5</td><td>S_{15}^5</td><td>S_{16}^5</td></tr>
</table>

Fig. 5. Original segmentations in BroadCatch (N = 2, 3, 4 and 5)

First consider positive channel transition from N to $N+1$ channels, which turns out to be permissible at any segment boundary. Fig. 6 shows an example of positive channel transition from 4 to 5 channels. At the point of transition, T_a, the payload on channel C_i, $1 \le i \le N$ switches from G_i^N to G_i^{N+1}. The switch, however, can be viewed as more nominal than actual because of (4). In other words, existing users will be able to receive exactly the same content they would receive from exactly the same channels. On the other hand, new viewers coming into the system at or after T_a can start receiving video using $N+1$ channels as if it is a new movie. The added channel, C_{N+1}, stays inactive for the length of one segment before transmitting G_{N+1}^{N+1} at time T_b, as specified in segment arrangement of BroadCatch. With the extra channel, the worst-case as well as average waiting time are reduced by 50%.

Negative channel transition from $N+1$ to N channels is similar to positive channel transition except that the transition point must coincide with segment boundary of S_i^N. Fig. 7 shows an example of negative channel transition from 5 to 4 channels. Again, because of (5), all but the releaseed channels deliver the same content before and after the point of transition, T_a. To ensure uninterrupted service, C_{N+1}, the channel to be released, needs to stop transmitting at time T_c. Consequently, new viewers have to wait until T_a before the viewing can start. For these users, the waiting time may be $L/(2^{N-2})$, which is equal to the worst case waiting time after the transition.

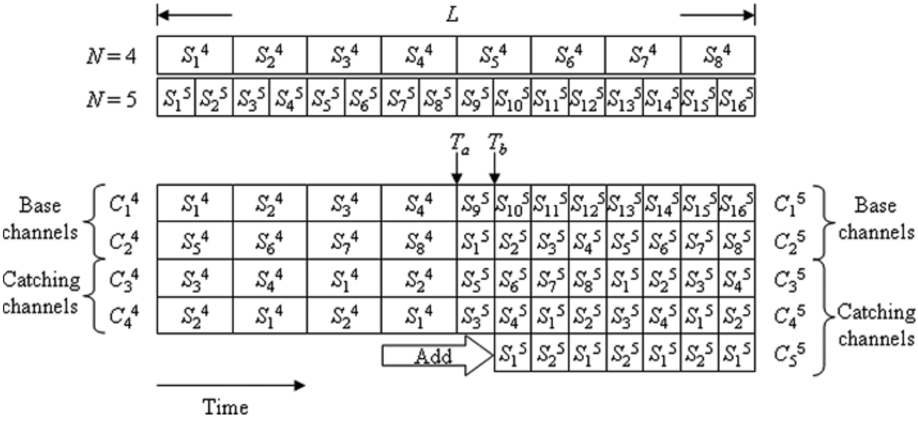

L
$N=4$: S_1^4 S_2^4 S_3^4 S_4^4 S_5^4 S_6^4 S_7^4 S_8^4
$N=5$: S_1^5 S_2^5 S_3^5 S_4^5 S_5^5 S_6^5 S_7^5 S_8^5 S_9^5 S_{10}^5 S_{11}^5 S_{12}^5 S_{13}^5 S_{14}^5 S_{15}^5 S_{16}^5
T_a T_b
Base channels
C_1^4: S_1^4 S_2^4 S_3^4 S_4^4 S_9^5 S_{10}^5 S_{11}^5 S_{12}^5 S_{13}^5 S_{14}^5 S_{15}^5 S_{16}^5 C_1^5
C_2^4: S_5^4 S_6^4 S_7^4 S_8^4 S_1^5 S_2^5 S_3^5 S_4^5 S_5^5 S_6^5 S_7^5 S_8^5 C_2^5
Catching channels
C_3^4: S_3^4 S_4^4 S_1^4 S_2^4 S_5^5 S_6^5 S_7^5 S_8^5 S_1^5 S_2^5 S_3^5 S_4^5 C_3^5
C_4^4: S_2^4 S_1^4 S_2^4 S_1^4 S_3^5 S_4^5 S_1^5 S_2^5 S_3^5 S_4^5 S_1^5 S_2^5 C_4^5
Add S_1^5 S_2^5 S_1^5 S_2^5 S_1^5 S_2^5 S_1^5 C_5^5
Base channels
Catching channels
Time

Fig. 6. Example of positive channel transition

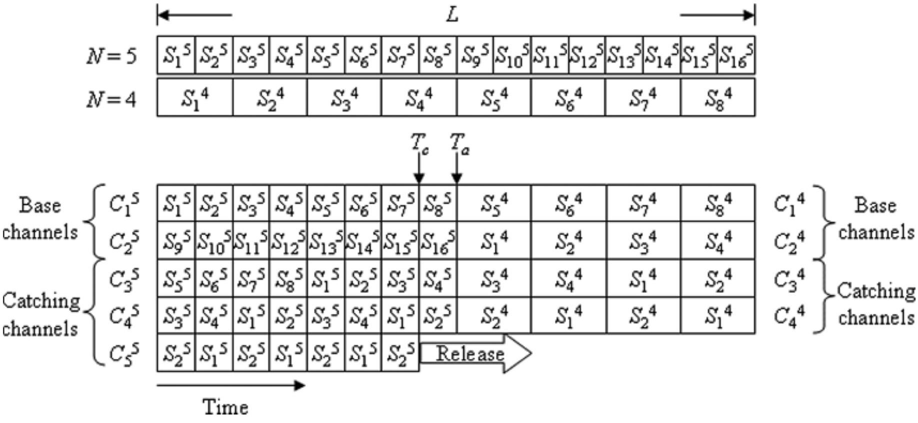

L
$N=5$: S_1^5 S_2^5 S_3^5 S_4^5 S_5^5 S_6^5 S_7^5 S_8^5 S_9^5 S_{10}^5 S_{11}^5 S_{12}^5 S_{13}^5 S_{14}^5 S_{15}^5 S_{16}^5
$N=4$: S_1^4 S_2^4 S_3^4 S_4^4 S_5^4 S_6^4 S_7^4 S_8^4
T_c T_a
Base channels
C_1^5: S_1^5 S_2^5 S_3^5 S_4^5 S_5^5 S_6^5 S_7^5 S_8^5 S_5^4 S_6^4 S_7^4 S_8^4 C_1^4
C_2^5: S_9^5 S_{10}^5 S_{11}^5 S_{12}^5 S_{13}^5 S_{14}^5 S_{15}^5 S_{16}^5 S_1^4 S_2^4 S_3^4 S_4^4 C_2^4
Catching channels
C_3^5: S_5^5 S_6^5 S_7^5 S_8^5 S_1^5 S_2^5 S_3^5 S_4^5 S_3^4 S_4^4 S_1^4 S_2^4 C_3^4
C_4^5: S_3^5 S_4^5 S_1^5 S_2^5 S_3^5 S_4^5 S_1^5 S_2^5 S_2^4 S_1^4 S_2^4 S_1^4 C_4^4
C_5^5: S_2^5 S_1^5 S_2^5 S_1^5 S_2^5 S_1^5 S_2^5 Release
Base channels
Catching channels
Time

Fig. 7. Example of negative channel transition

3.2 Comparison

Our proposed channel transition for BroadCatch has two advantages over the one presented in section 2.2. First, although both schemes are seamless, ours is more seamless in the sense that the channel payload is not affected in any way by a transition. This makes our scheme easier to understand and to implement. Second, BroadCatch clients receive data from at most $N-2$ out of N channels. Assuming bandwidth is not a major concern at the server and intermediate routers, we can justifiably compare seamless channel transition on fast broadcasting using N channels against our scheme using $N+2$ channels, as both require the same number of channels on the client side. The worst-case waiting time in our scheme is $L/2^{N+1}$, which is 62.5% shorter than $L/(3\times 2^{N-2})$, the worst-case waiting time in [8].

4 Simulation Results

The main purpose of channel transition is to give a multi-movie Video-on-Demand system the ability to dynamically reallocate channels based on movie popularity. In this paper, we simulate a BroadCatch-based VoD system with the following parameters. Let m be the number of movies served by the system and each movie is allocated i channels initially. Channels may be added to or taken away from a movie depending on its popularity, but the number of allocated channels stays within an upper bound, u , and a lower bound, l .

The decision as to when a channel transition should take place is governed by the following strategy. If the number of requests for movie A during the previous segment is α times smaller that for another movie B, then the popularity of movie A is considered less popular and one channel will be taken away from A and reallocated to B.

Variation of movie popularity is modeled by a Zipf distribution [12,13]:

$$P(X=k)=\begin{cases}\dfrac{(1/k)^{\theta}}{\sum_{i=1}^{n}(1/i)^{\theta}} & ,k=1,\ldots,n\\ 0 & ,otherwise\end{cases} \tag{6}$$

where the parameter θ is the skew factor. When θ is 0, the Zipf distribution is reduced to a uniform distribution, meaning that all movies have the same access frequency. On the other hand, if θ is large, then the access frequencies of some movies are higher than that of others.

The arrival of client request is modeled by a Poisson distribution

$$P(X=k)=\frac{e^{-\lambda}\lambda^{k}}{k!}, \tag{7}$$

where λ is the client request rate.

Each movie is assumed to be 120 minutes long. The simulation time is 24 hours. Video-on-Demand systems of 10, 15 and 20 movies are simulated. Other parameters are assigned fixed values as follows: $i=5$, $\alpha=8$, and $\lambda=5$ requests per minute.

Fig. 8 shows the average waiting time for various choices of u , l and m . The skew factor θ in the Zipf distribution is set to 0.729, as suggested in [12]. The leftmost sample points in Fig. 8. represent the traditional BroadCatch method with $N=5$. As can be seen in Fig. 8, enabling channel transition reduces the average waiting time by more than 10%.

Fig. 9 plots the average waiting time for the tradition BroadCatch method against the proposed method for skew factor θ between 0 to 2. It shows that the average waiting time for the proposed method, which adapts itself to non-uniform movie popularity, is a decreasing function of θ. For $\theta>0.5$, the proposed method has a

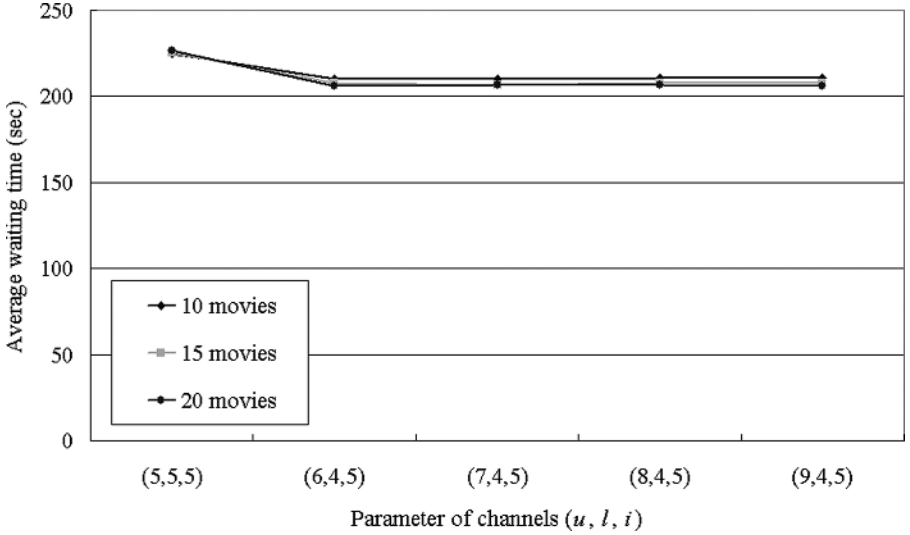


Fig. 8. Average waiting time on a BroadCatch-based VoD system ($\theta = 0.729$)

smaller average waiting time than that for BroadCatch. Fig. 10 shows the average number of allocated channels and the number of total viewers for the 20 movies in the VoD system. It shows that more channels are indeed allocated to more popular movies, leading to the reduction of average waiting time.

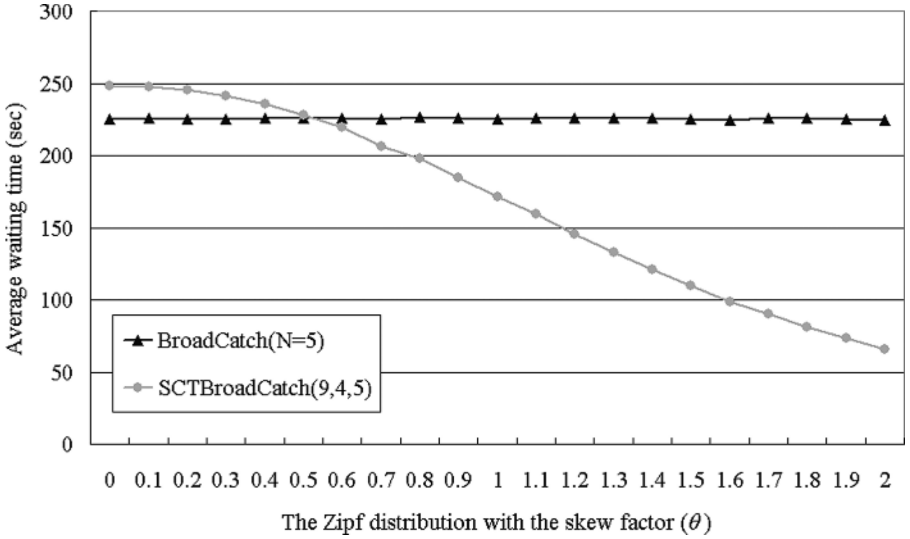


Fig. 9. Average waiting time as a function of the skew factor θ (20 movies)

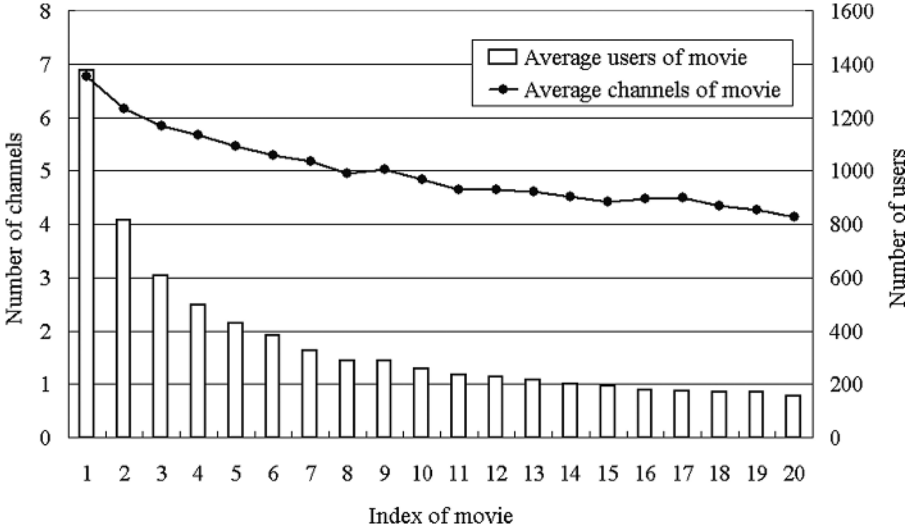


Fig. 10. Average number of allocated channels and the number of users ($\theta = 0.729$, $(u, l, i) = (9,4,5)$)

5 Conclusion

Various broadcasting schemes are used in Near Video-on-Demand systems to serve multiple movies to a large number of viewers. In these systems, seamless channel transition provides the ability to dynamically reallocate channels without service interruption and thus can be used as an important performance tuning tool. In this paper, we show how seamless channel transition can be achieved for the BroadCatch scheme. Our simulation shows that the average waiting time can be reduced when seamless channel transition is employed along with a suitable transition strategy.

Acknowledgments. This research is supported in part by an ISU grant, ISU-94-01-08.

References

1. H. Shachnai and P. S. Yu, "The role of wait tolerance in effective batching: A paradigm for multimedia scheduling schemes," *Technical Report RC 20038*, IBM Research Division, T.J. Watson Research Center, Apr. 1995.
2. L-S. Juhn and L-M. Tseng, "Fast broadcasting for hot video access," *RTCSA'97: the proceedings of the 4th international workshop on real-time computing system and applications*, pp.237-243, Oct. 1997.
3. S. Viswanathan and T. Imielinski, "Metropolitan area video on demand service using pyramid broadcasting," *Multimedia System*, vol.4, no.4, pp.197-208, Aug. 1996.
4. K.A. Hua and S. Sheu, "Skyscraper broadcasting: a new broadcasting scheme for metropolitan video on demand system," *ACM SLGCOMM97*, Cannes, France, pp.89-100, Sep. 1997.

5. L.S. Juhn and L. M. Tseng, “Staircase data broadcasting and receiving scheme for hot video service,” *IEEE Transaction on Consumer Electronics*, vol.43, no.4, pp.1110-1117, Nov. 1997.
6. L. S. Juhn and L. M. Tseng, “Harmonic broadcasting for video on demand service,” *IEEE Transaction on Broadcasting*, vol.43, no.3, pp.268-271, Sep. 1997.
7. M.A. Tantaoui, K.A. Hua, and T.T. Do, “BroadCatch: a periodic broadcast technique for heterogeneous video-on-demand,” *IEEE Transactions on Broadcasting*, vol.50, no.3, pp.289-301, Sep. 2004.
8. Yu-Chee Tseng, Ming-Hour Yang, Chi-Ming Hsieh, Wen-Hwa Liao, and Jang-Ping Sheu, “Data broadcasting and seamless channel transition for highly demanded videos,” *IEEE Transactions on Communications*, vol.49, no.5, pp.863-874, May 2001.
9. Yukon Chang and Shou-Li Hsu, “A flexible architecture for unified Video-on-Demand Services,” *Workshop on Consumer Electronics and Signal Processing(WCEsp2004)*, Hsinchu, Taiwan, Nov. 2004.
10. Yu-Chee Tseng, Yu-Chi Chueh, and J.P. Sheu, “Seamless channel transition for the staircase video broadcasting scheme,” *IEEE/ACM Transactions on Networking*, vol.12, pp.559-571, Jun. 2004.
11. Wei-De Chien, Yuan-Shiang Yeh, and Jia-Shung Wang, “Practical channel transition for Near-VoD Services,” *IEEE Transactions on Broadcasting*, vol.51, pp.360-365, Sep. 2005.
12. Asit Dan, Dinkar Sitaram, and Perwez Shahabuddin, “Scheduling Policies for an On-Domand Video Server with Batching,” *ACM Multimedia conference*, pp.15-23, 1994.
13. S. Ramesh, I. Rhee, and K. Guo, “Multicast with Cache (Mcache): An Adaptive Zero-Delay Video-on-Demand Service,” *IEEE Infocom 2001*, vol.1, pp.85-94, 2001.

Spatial Interpolation Algorithm for Consecutive Block Error Using the Just-Noticeable-Distortion Method

Su-Yeol Jeon, Chae-Bong Sohn, Ho-Chong Park, Chang-Beom Ahn, and Seoung-Jun Oh

VIA-Multimedia Center, Kwangwoon University, 447-1, Wolgye-Dong, Nowon-Gu,
139-701, Seoul, Korea
{k2ambo, via-staff}@viame.re.kr

Abstract. This paper presents a new error concealment algorithm based on edge-oriented spatial interpolation as well as the JND (Just-Noticeable-Distortion) function to select valid edge information in adjacent blocks. Several conventional algorithms suffer from computational complexity or inaccurate recovery at missing blocks having strong edgeness. In order to alleviate these drawbacks, the proposed algorithm estimates a dominant direction to interpolate the missing pixel values. At this step, automatic thresholding method based on the JND function is proposed to determine a valid direction. Then, a lost block is recovered by weighted linear interpolation along the dominant vector. Finally, the median filter is applied to recover pixels that are not located on the line with the estimated dominant direction. Simulation results show that the proposed algorithm improves the PSNR of the recovered image approximately 2.5dB better than Hsia's and can significantly reduce computational complexity with keeping subjective quality as good as the RIBMAP method.

Keywords: Error concealment, block-loss recovery, JND function, matching vector, interpolation.

1 Introduction

The block-loss recovery is very important for block-based coding, such as JPEG and MPEG standards. When a block is lost, it can be estimated by using adjacent pixels in post processing. Many spatial interpolation techniques for block-loss recovery have been proposed. There is no acceptable algorithm in terms of computational complexity as well as recovery accuracy for commercial products.

Lee *et* al. [1] introduced a cubic spline interpolation using sliding recovery window. Tsekeridou and Pitas [2] proposed a split-match algorithm using 'best match' 4x4 blocks. Sun and Kwok [3] presented a spatial interpolation algorithm using projections onto convex sets [4].

Among the spatial domain error concealment algorithms, the Hsia's algorithm achieves relatively good quality, compared to the other algorithms with low computational complexity [5]. This algorithm recovers the error blocks by identifying two 1-D boundary matching vectors. However, its recovery direction is still

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 979 – 987, 2006.

inaccurate since edge directions of boundary pixels are not considered. Especially, the performance of Hsia's algorithm is declined for the regions having multiple edge directions in boundary pixels.

In cope with this shortcoming, we propose a new error recovery method of which the key operation is to efficiently find a boundary matching vector according to neighboring block edges. This scheme is organized as follows. Firstly, the proposed algorithm detects edges with neighboring blocks and selects valid directions based on the JND function [6]. Then, a dominant direction is selected as the matching vector among detected valid edges. Finally, according to the direction of the matching vector, spatial interpolation is applied to lost blocks.

The rest of this paper is organized as follows. In Section 2, the proposed algorithm is presented in detail. Simulation results are shown in Section 3. Finally, Section 4 concludes this paper.

2 Proposed Algorithm

2.1 Adjacent and Connected Blocks

A lost block, *L*, and its surrounding neighboring blocks are illustrated in Fig. 1. In Fig. 1(a), Six neighboring blocks, A_{TL}, A_{T}, A_{TR}, A_{BL}, A_{B} and A_{BR}, and boundary pixel lines, *D*, are shown with the lost block, *L*, in the center, of which size is *N*×*N*. Fig. 1(b) shows 3x3 size boundary blocks, *B*, which are adjacent to the lost block and are in neighboring blocks of the lost block, *L*.

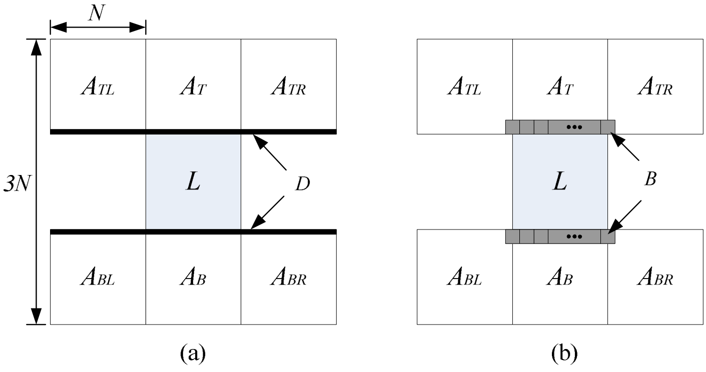


Fig. 1. Lost block and its neighboring blocks (a) Lost block *L* and six neighboring blocks *ATL*, *AT*, *ATR*, *ABL*, *AB* and *ABR*. (b) Lost block *L* and boundary blocks, *B*.

In the edge-oriented spatial interpolation algorithm proposed by Hsia, lost blocks are restored with 1-D boundary matching between top and bottom boundary pixel lines in the neighboring blocks. Since this algorithm does not count on the edge directions of adjacent blocks, it can deteriorate as shown in Fig. 2.

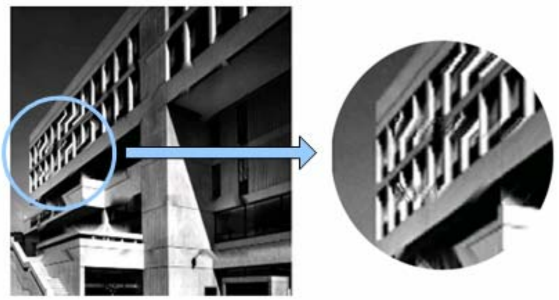


Fig. 2. The recovered image using Hsia's spatial interpolation algorithm for MIT image

2.2 JND Function-Based Edge Detection

To find the edge direction and magnitude for adjacent blocks, *A*, sobel edge detector in the spatial domain is applied to 3x3 boundary blocks, *B*. As shown in Fig. 3, each boundary block is divided into two regions according to the edge direction. If the difference between two divided regions is larger than the JND value at the center of two regions, the edge is selected as a valid edge. Note that the JND represents the smallest difference in a specified modality of sensory input that is detectable by human being.

$$JND(g(x,y)) = \begin{cases} T_o(1-\sqrt{g(x,y)/127}+3), & for\ g(x,y) \le 127 \\ \gamma(g(x,y)-127)+3, & otherwise \end{cases} \tag{1}$$

where $g(x, y)$ is a gray scale value at the position (x, y), $T_o = 17$ and $\gamma = 3/128$ [6].

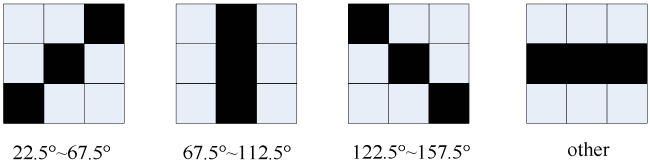


Fig. 3. Edge direction patterns for 3x3 boundary block, *B*

2.3 Matching Vector Selection

Matching vector selection is depicted in Fig. 4. According the JND function-based edge detection, each boundary block may have valid edge direction which is the candidate of the best matching vector. The best matching vector is selected as one direction that are most frequently found in all the valid directions. The selected best matching vector represents dominant edge direction of neighboring blocks.

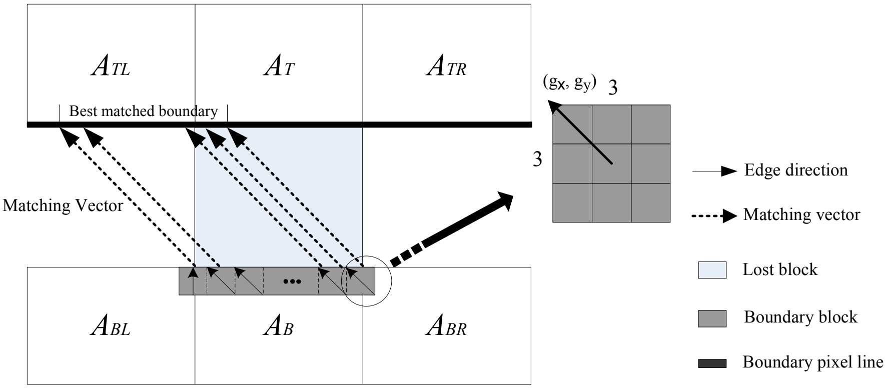


Fig. 4. Matching vector selection

This selecting process is executed on the top and bottom boundary block, *B*, respectively. As a result, each lost block, *L*, has two or less number of matching vectors.

2.4 Spatial Interpolation

To recover block losses, spatial interpolation is employed to each lost block, *L*. And, lost block restoration is accomplished by the linear spatial interpolation of which recovery direction is set to the direction of the matching vector.

Fig. 5 shows the proposed spatial interpolation algorithm with two matching vectors. To find more dominant matching vectors between top and bottom lines, 1-D block boundary matching is employed. The pixel is interpolated along the dominant matching vector as shown in Fig. 5(a). Then, unrecovered pixels are interpolated along another matching vector as shown in Fig. 5(b).

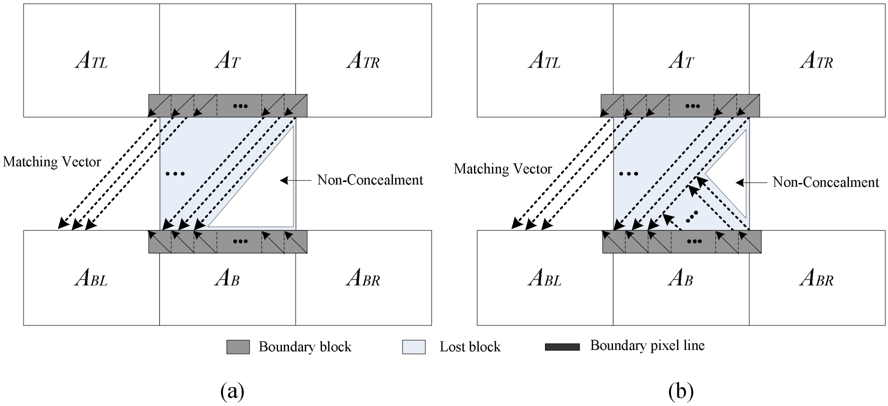


Fig. 5. An example of proposed spatial interpolation algorithm with two-matching vector case

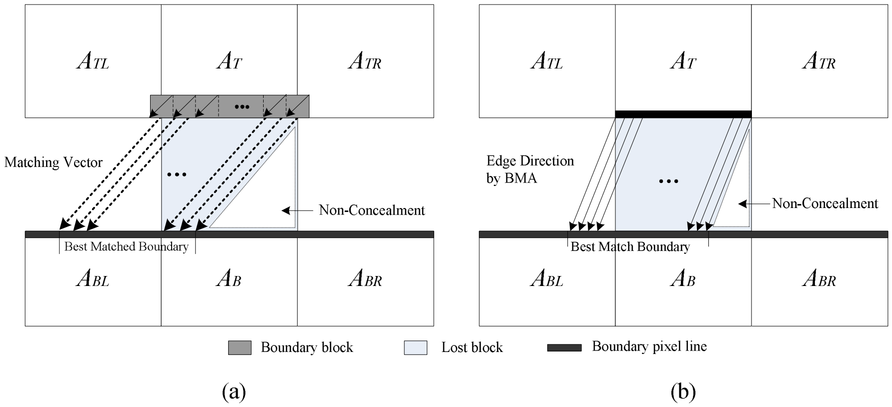


Fig. 6. An example of the proposed spatial interpolation algorithm (a) Single matching vector case (b) No matching vector case

According to the proposed edge detection rule, lost blocks do not always have two matching vectors. Fig. 6 shows the case that a lost block does not have two matching vectors. Fig. 6(a) depicts the single matching vector case and Fig. 6(b) shows the no matching vector case that a lost block does not have any matching vectors. In these cases, non concealment region can be found at a lost block. Accordingly, 1-D block boundary matching to find the edge direction is used to determine which direction is more accurate[5]. This rule is applied to top and bottom directions, respectively. Fig. 7 shows that a few pixels are not recovered even after two spatial interpolations. Pixels in non-concealment areas are replaced with the median filtered values.

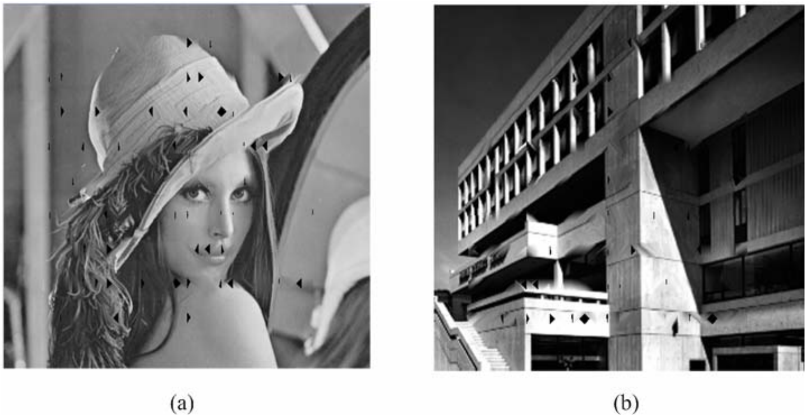
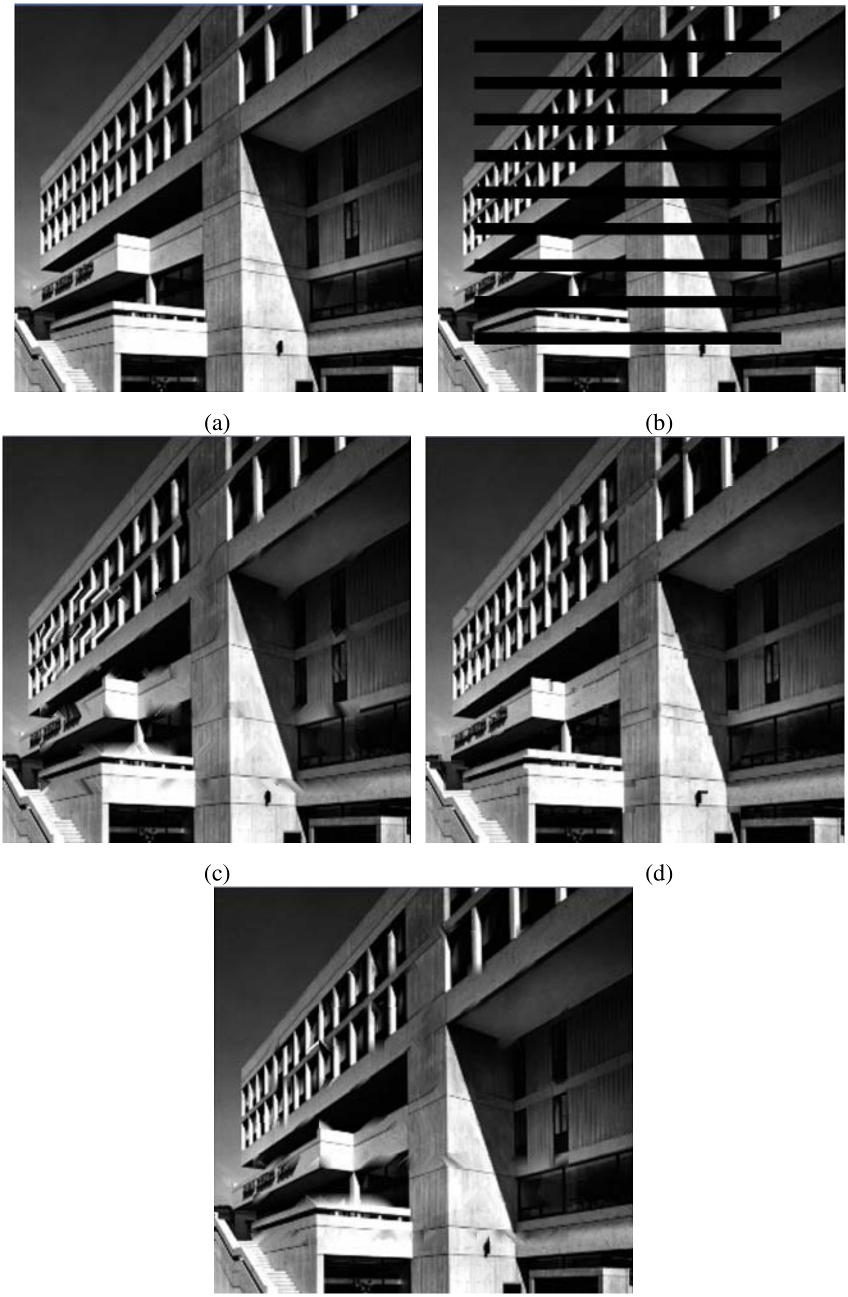


Fig. 7. Examples of spatial interpolated image. A small number of pixels are not recovered. (a) Lena (b) MIT.



(a)

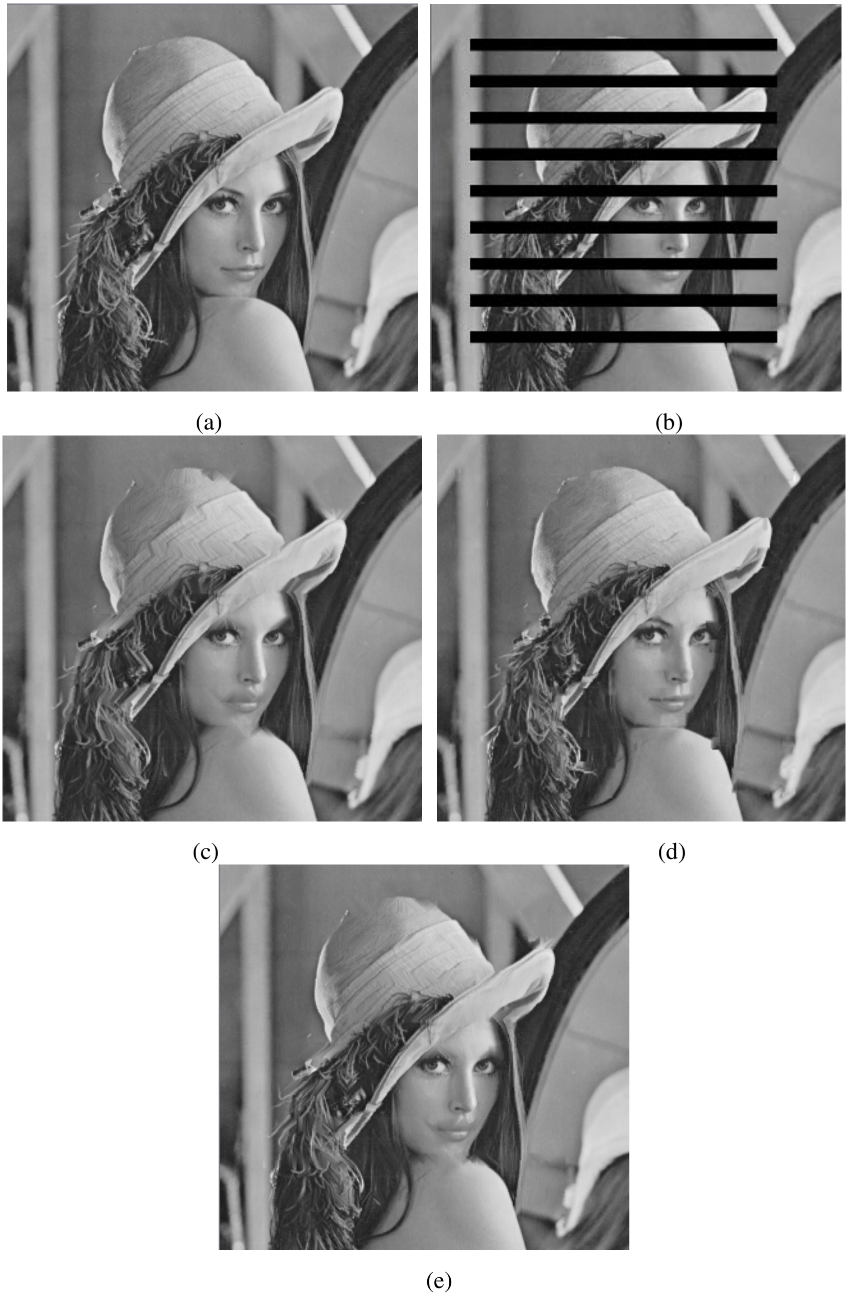
(b)

(c)

(d)

(e)

Fig. 8. Test results for MIT: (a) Original image (b) Damaged image (c) Hsia's (d) RIBMAP and (e) Proposed algorithm



(a)

(b)

(c)

(d)

(e)

Fig. 9. Test results for Lena: (a) Original image (b) Damaged image (c) Hsia's (d) RIBMAP and (e) Proposed algorithm

3 Experiments and Results

The performance of the proposed algorithm is evaluated with two well-known 512×512 test images: MIT and Lena. The proposed algorithm is compared with Hsia's [5] and Park's [4] in terms of PSNR. The PSNRs of the proposed and the conventional algorithms are summarized in Table 1. According to Table 1, the proposed algorithm provides a slightly better PSNR value than Hsia's in case of Lena image. However, in case of MIT image, the PSNR of the proposed algorithm is improved approximately 2.5 dB, compared with Hsia's one.

Table 1. Comparison of PSNRs for different error concealment algorithms

Algorithms	MIT	Lena
Hsia's scheme	23.784	29.735
RIBMAP	28.127	31.706
Proposed scheme	26.262	29.958

The subjective qualities of Hsia's, Park's [4] and the proposed algorithm were evaluated. Park's RIBMAP algorithm shows the best restoration performance in terms of the subjective and objective qualities. However, this algorithm requires high computational complexity. N recovery vector searches and additional filtering are required for N×N lost block restoration in the RIBMAP algorithm. Fig. 8 and 9 show that the proposed algorithm yields significant improvement in terms of subjective quality, compared with Hsia's and comparable subjective quality with the RIBMAP. According to these test results, we found that the proposed algorithm is better than Hsia's one in terms of objective and subjective cases while the proposed algorithm yields similar image quality with low computational complexity, compared with the RIBMAP.

4 Conclusion

The error correction for both still images and intra-frames is one of key techniques for block-based video coding. Hsia proposed the efficient algorithm for concealment of damaged blocks using boundary matching vector. However, Hsia's algorithm could be degraded when there are several edges in boundary pixels.

In this work, we proposed the new efficient spatial interpolation algorithm for block error concealment based on the JND-oriented edge direction matching. In this algorithm, the best matching vector is selected as the most dominant edge direction in boundary blocks according the JND function. Experimental results show that the proposed scheme yields around 2.5 dB better than Hsia's one in terms of PSNR. With the low computational complexity, the image quality recovered by the proposed algorithm is similar to that of the RIBMAP.

References

1. X. Lee, Y.-Q. Zhang, and A. Leon-Garcia, "Information loss recovery for block-based image coding techniques-a fuzzy logic approach," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 259–273, Mar. 1995.
2. S. Tsekeridou and I. Pitas, "MPEG-2 error concealment based on block matching principles," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 4, pp. 646–658, Jun. 2000.
3. H. Sun and W.Kwok, "Concealment of damaged blocks transform coded images using projections onto convex sets," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 470–477, Apr. 1995.
4. J. Park, D. Park, R. Marks, and M. El-Sharkawi, "Recovery of Image Blocks Using the Method of Alternating Projections," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 461-474, Apr. 2005.
5. S. C. Hsia, "An edge-oriented spatial interpolation for consecutive block error concealment," *IEEE Signal processing letters*, vol.11, no. 6, pp. 577–580, June 2004.
6. C. H. Chou and Y. C. Li, "A perceptually Tuned Subband Image Coder Based on the Measure of Just-Noticeable-Distortion Profile," *IEEE Trans. Circuits Syst., Video Technol.*, vol. 5, no. 6, Dec. 1995.

Error Resilient Motion Estimation for Video Coding

Wen-Nung Lie[1], Zhi-Wei Gao[1], Wei-Chih Chen[1], and Ping-Chang Jui[2]

[1] Department of Electrical Engineering
National Chung Cheng University, Chia-Yi, 621, Taiwan, ROC
wnlie@ee.ccu.edu.tw, cwkao@samlab.ee.ccu.edu.tw,
u9142042@ccu.edu.tw
[2] Materials & Electro-Optics Research Division, Chung Shan Institute of Science &
Technology, Lung-Tan, Tao-Yuan 325, Taiwan, ROC

Abstract. Due to temporal prediction adopted in most video coding standards, errors occurred in the current frame will propagate to succeeding frames that refer to it. This causes substantial degradation in reconstructed video quality at decoder side. In order to enhance robustness of existing temporal prediction techniques, another prediction strategy, called error resilient motion estimation (ERME), to take both coding efficiency and error propagation into considerations is proposed. To find MVs that satisfy the above two requirements, a constrained optimization problem with a new criterion is thus formed for ME. The proposed algorithm is implemented for H.264 video coding standard, where multiple reference frames are allowed for ME. From experimental results, the proposed algorithm can improve PSNR by up to 1.0 dB (at a packet loss rate of 20%) when compared with the full search ME with traditional SAD (sum of absolute difference) criterion.

Keywords: H.264, Error resilience, Error concealment, Motion estimation.

1 Introduction

Due to the successful development of techniques, such as Discrete Cosine Transform (DCT), Motion Estimation and Compensation (ME/MC), and Variable Length Coding (VLC), a large amount of redundant information in video can be removed and allow the data size to be reduced for transmission over channels of limited bandwidth. However, compressed videos are vulnerable to channel errors; even a minor disturbance may make the received video bit stream undecodable. Hence, techniques, such as error resilient video coding and error concealment [1], [2], were proposed to enhance the robustness of video against errors or to recover videos from errors.

To suppress error propagation without sacrificing coding efficiency, a method of searching motion vectors (MVs) in a criterion other than least SAD (sum of absolute difference) was proposed [3], [4]. This kind of algorithms could be classified as Error Resilient Motion Estimation (ERME). Methods proposed in [3], [4] adopt the end-to-end distortion as a new criterion for motion search. Although the above-referred methods are capable of suppressing error propagation efficiently, their computational complexity in estimating the end-to-end distortion is prohibitively high and thus impractical for real-time applications.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 988 – 996, 2006.

Extending the prior concept to multiple reference frames (up to 5) for ME in AVC/H.264 video coding standard [7] and avoiding the inherited high computational complexity in estimating the end-to-end distortion, a new ERME algorithm is proposed in this paper. After decomposing the end-to-end distortion into parts of source and channel distortions, an objective measure, which can be estimated with much lower computational complexity, for evaluating the potential error propagation of an MV is derived in this paper. By integrating this objective measure into the conventional search criterion for ME, a constrained optimization problem to derive error resilient MVs can be formed. Importantly, we consider not only error resilience, but also coding efficiency of the estimated MVs, that is, tradeoffs will be made between them in presence of channel errors.

The main contribution of this work is that an ERME algorithm of low complexity is developed. Experiments show that our proposed algorithm outperforms the traditional ME method of AVC/H.264 by improving the PSNR by up to 1.0 dB in case of a packet loss rate of 20%.

2 Concept of Error Resilient Motion Estimation

The goal of motion estimation is to enhance the coding efficiency by removing the temporal correlation among frames. However, in error-prone environments, it is known that error propagations might be induced in received videos due to incomplete error concealment and the process of motion compensation (MC). In order to compromise between the suppression of error propagation and the retaining of coding efficiency, the said error resilient motion estimation was then proposed [3], [4].

In [3], a criterion that considers end-to-end distortion, that is, the distortion between the original input frame and the decoded frames at the receiver end, for motion estimation was proposed. Then, MVs that can minimize the defined criterion is chosen for encoding. Their experimental results show better qualities with respect to traditional MVs. In [4], a similar motion estimation algorithm as [3] was proposed, except that stochastic frame buffers were used as references rather than conventional reconstructed frame buffers. Although their experimental results show that the use of stochastic frame buffers can improve qualities of the decoded videos, their algorithm is not standard-compatible. A common drawback of the criterion used in [3] and [4] for motion estimation is the extra high computational cost that is induced.

The criterion proposed in [3] separates the errors into three sources: error propagation, error concealment, and quantization. To evaluate the distortion from quantization, annoying computational cost arises since DCT/IDCT and quantization should be incorporated into motion estimation process. Here in this paper, we use a simplified criterion, which is capable of retaining the error resilience performance, so that ERME can be feasible in practical applications.

On the other hand, the concept of minimizing end-to-end distortions is similarly applied to optimally deciding the coding mode (intra or inter) for each macroblack (MB) [8], [9], [10]. However, their MVs are estimated based on the traditional SAD criterion before the mode decision is optimally made based on an estimated end-to-end distortion. Theoretically, our algorithm for error resilient motion estimation can be used for finding MVs for mode decision based again on the end-to-end distortion.

Before illustrating our proposed algorithm, we first model the end-to-end distortion for videos transmitted in error-prone environments as follows [5]: end-to-end distortion can be decomposed into parts of source distortion D_s, incurred by n_q, and channel distortion D_c, incurred by n_c. Here, n_q is related to the quantization noise and n_c to the incompleteness of error concealment and motion compensation. A pictorial illustration of our model is depicted in Fig. 1, where f_n represents the original video signal, $\hat{f}_n$ represents the encoded video (i.e., decoded video without error), and $\tilde{f}_n$ represents the video reconstructed at the decoder side in presence of channel noise n_c.

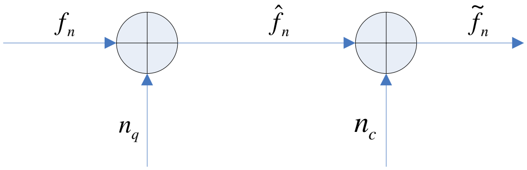


Fig. 1. The modeling of end-to-end distortion

Via the above model, it is ready to observe that the end-to-end distortion $D_{end\text{-}to\text{-}end}$ can be formulated as

$$\begin{aligned} D_{end-to-end} &= E\left\{(f_n - \tilde{f}_n)^2\right\} \\ &\cong E\left\{(f_n - \hat{f}_n)^2 + (\hat{f}_n - \tilde{f}_n)^2\right\} = D_s + D_c \end{aligned}, \tag{1}$$

where $D_s = E\left\{(f_n - \hat{f}_n)^2\right\}$ and $D_c = E\left\{(\hat{f}_n - \tilde{f}_n)^2\right\}$.

This model was first verified in [5] based on a test platform using H.263 video coding standard. In order to verify its applicability to H.264/AVC standard, the following process is conducted. In Eq. (2), a measure of deviation e_D is defined

$$e_D = \frac{1}{N}\sum_{n=1}^{N}\frac{\left|\left[D_S(n) + D_C(n)\right] - D_{end-to-end}(n)\right|}{D_{end-to-end}(n)} \times 100\%\,, \tag{2}$$

where n is the frame index and N is the total number of frame in a video sequence.

Simulations were made by encoding videos, dropping packets randomly in terms of indicated rates, recovering the lost MBs, and measuring D_s, D_c, and $D_{end\text{-}to\text{-}end}$, respectively. After a set of experiments on 4 video sequences under varying packet loss rate (PLR), it was observed that simplification about the end-to-end distortion is sustained by an averaged e_D of only 0.863%. Based on this fact, $D_{end\text{-}to\text{-}end}$ can then be reduced by minimizing D_s and D_c separately.

To derive error resilient MVs, the channel distortion D_c should be modeled first. The search criterion is defined below:

$$(\Delta x^*, \Delta y^*) = \arg\min_{(\Delta x, \Delta y)\in S}\left\{\sum_{i=1}^{p} E\left\{\left(\hat{f}_n^{\,i}(\Delta x, \Delta y) - \tilde{f}_n^{\,i}(\Delta x, \Delta y)\right)^2\right\}\right\}, \tag{3}$$

where Δx and Δy are the horizontal and vertical components of the MV, S is the feasible set for motion vectors, p is the number of pixels in a block, $\hat{f}_n^i(\Delta x, \Delta y)$ and $\tilde{f}_n^i(\Delta x, \Delta y)$ represent the i^{th} pixel of a block motion-compensated via MV= $(\Delta x, \Delta y)$ in the n^{th} frame, and $E\{.\}$ is the expectation operator. Notice that $\hat{f}_n^i(\Delta x, \Delta y)$ represents the pixel value correctly decoded, while $\tilde{f}_n^i(\Delta x, \Delta y)$ is the pixel value considering erroneous reconstruction.

Recognizing that most video coding standards adopt SAD as the criterion for motion search, a new criterion below is formed by changing the squared term in Eq.(3) with an absolute term:

$$(\Delta x^*, \Delta y^*) = \arg \min_{(\Delta x, \Delta y) \in S} \left\{ \sum_{i=1}^{p} E\left\{ \left| \hat{f}_n^i(\Delta x, \Delta y) - \tilde{f}_n^i(\Delta x, \Delta y) \right| \right\} \right\}. \tag{4}$$

Based on the fact that $E\{|\hat{x} - \tilde{x}|\} \geq |E\{\hat{x} - \tilde{x}\}|$, the following inequality will hold.

$$\begin{aligned}(\Delta x^*, \Delta y^*) &= \arg \min_{(\Delta x, \Delta y) \in S} \left\{ \sum_{i=1}^{p} E\left\{ \left| \hat{f}_n^i(\Delta x, \Delta y) - \tilde{f}_n^i(\Delta x, \Delta y) \right| \right\} \right\} \\ &\geq \arg \min_{(\Delta x, \Delta y) \in S} \left\{ \sum_{i=1}^{p} \left| E\left\{ \hat{f}_n^i(\Delta x, \Delta y) - \tilde{f}_n^i(\Delta x, \Delta y) \right\} \right| \right\} \\ &= \arg \min_{(\Delta x, \Delta y) \in S} \left\{ \sum_{i=1}^{p} \left| \hat{f}_n^i(\Delta x, \Delta y) - E\left\{ \tilde{f}_n^i(\Delta x, \Delta y) \right\} \right| \right\}\end{aligned} \tag{5}$$

Note that the first term in the summation is considered to be deterministic with respect to the $E\{.\}$ operator, due to its independence to the channel conditions. In Eq.(5), $E\{\tilde{f}_n^i(\Delta x, \Delta y)\}$ can be estimated easily by using formula blow, which was derived in [6].

$$\begin{aligned} E[\tilde{f}_n^i(\Delta x, \Delta y)] = (1 - p_e)\{ & E[\tilde{f}_{n-\alpha}^{i,j}(\Delta x, \Delta y)] \\ & + r_n^i(\Delta x, \Delta y)]\} + p_e \cdot E[\tilde{f}_{n-1}^i] \end{aligned}, \tag{6}$$

where p_e is the error probability of a considered pixel, $r_n^i(\Delta x, \Delta y)$ is the residual produced after motion compensation, $E[\tilde{f}_{n-\alpha}^{i,j}(\Delta x, \Delta y)]$ is the pixel value compensated from the j^{th} pixel of the $(n\text{-}\alpha)^{\text{th}}$ frame, and $E[\tilde{f}_{n-1}^i]$ is the recovered pixel value if the zero motion scheme is adopted for error concealment.

According to coding principle, $\hat{f}_n^i(\Delta x, \Delta y)$ can be represented as:

$$\hat{f}_n^i(\Delta x, \Delta y) = \hat{f}_{n-\alpha}^{i,j}(\Delta x, \Delta y) + r_n^i(\Delta x, \Delta y) \tag{7}$$

where $\hat{f}_{n-\alpha}^{i,j}(\Delta x, \Delta y)$ is the motion-compensated pixel value from the $(n\text{- })^{\text{th}}$ frame. Substituting Eq.(6) and (7) into Eq.(5), we obtain:

$$(\Delta x^*,\Delta y^*) = \arg\min_{(\Delta x,\Delta y)\in S}\left\{\sum_{i=1}^{p}\left|\begin{array}{l}\left(\hat{f}^{i,j}_{n-\alpha}(\Delta x,\Delta y)+r^i_n(\Delta x,\Delta y)\right)-p_e E[\hat{f}^i_{n-1}]\\ -(1-p_e)\left(E\left[\tilde{f}^{i,j}_{n-\alpha}(\Delta x,\Delta y)\right]+r^i_n(\Delta x,\Delta y)\right)\end{array}\right|\right\} \tag{8}$$

Eq.(8) is actually not a good criterion, due to its prohibitively high complexity in computing $r^i_n(\Delta x,\Delta y)$. To yield $r^i_n(\Delta x,\Delta y)$, the processes of DCT, quantization, and inverse DCT need to be performed for each MV candidate. This is also the reason why algorithms provided in [3], [4] are impractical in terms of computing complexity.

The above equation can be rewritten as

$$(\Delta x^*,\Delta y^*) = \arg\min_{(\Delta x,\Delta y)\in S}\left\{\sum_{i=1}^{p}\left|\begin{array}{l}\left(\hat{f}^{i,j}_{n-\alpha}(\Delta x,\Delta y)\right)-\left((1-p_e)\cdot E\left[\tilde{f}^{i,j}_{n-\alpha}(\Delta x,\Delta y)\right]+p_e\cdot E[\hat{f}^i_{n-1}]\right)\\ +\left(p_e\cdot r^i_n(\Delta x,\Delta y)\right)\end{array}\right|\right\} \tag{9}$$

The term $p_e \cdot r^i_n(\Delta x,\Delta y)$ can be ignored, with respect to the other three terms, to reduce Eq.(9) into Eq.(10).

$$(\Delta x^*,\Delta y^*) = \arg\min_{(\Delta x,\Delta y)\in S}\left\{\sum_{i=1}^{p}\left|\left(\hat{f}^{i,j}_{n-\alpha}(\Delta x,\Delta y)\right)-\left((1-p_e)\cdot E\left[\tilde{f}^{i,j}_{n-\alpha}(\Delta x,\Delta y)\right]+p_e\cdot E[\hat{f}^i_{n-1}]\right)\right|\right\} \tag{10}$$

The reasons are stated below. In Eq.(9), the term $r^i_n(\Delta x,\Delta y)$ is the prediction error of a certain MV, which is often small if high correlations exist between consecutive video frames. However, the second and third terms in Eq.(9), i.e., $E[\tilde{f}^{i,j}_{n-\alpha}(\Delta x,\Delta y)]$ and $E[\tilde{f}^i_{n-1}]$, represent the expected reconstructions at decoder and normally cannot be ignored. By the way, we consider a value of p_e of no more than 0.3 in our system.

In Eq.(10), the first term $\hat{f}^{i,j}_{n-\alpha}(\Delta x,\Delta y)$ is readily available when encoding the n^{th} frame, while the second term (the first moment of $\tilde{f}^{j}_{n-\alpha}$) can be derived by using the technique proposed in [6]. Note from Eq.(6) that the first moment of $\tilde{f}^i_n$ can be recursively updated for next frame's use *after* a frame is encoded (i.e., after the residual r^i_n is figured out), hence guaranteeing the availability of $E[\tilde{f}^{i,j}_{n-\alpha}(\Delta x,\Delta y)]$ on evaluating Eq.(10) for the n^{th} frame. Clearly, the extra computations required in evaluating Eq.(10) come from the updating of $E[\tilde{f}^i_{n-1}]$.

The criterion proposed in Eq.(10) totally ignores the coding efficiency in optimization. It is possible that MVs and residuals thus obtained consume a lot of bit rates for transmission, though a certain degree of robustness can be achieved. Before proceeding, some mathematical terms are defined. Denote $(\boldsymbol{w},\alpha)$ to be a MV candidate $\boldsymbol{w}$ that refers to the (n-)th frame. Define $EP_{(\boldsymbol{w},\alpha)}$ and $CE_{(\boldsymbol{w},\alpha)}$ to measure the level of error propagation and coding efficiency, respectively, for a given $(\boldsymbol{w},\alpha)$, as follows.

$$EP_{(\boldsymbol{w},\alpha)} = \sum_{i=1}^{p} \left| \hat{f}_{n-\alpha}^{i,j}(\boldsymbol{w},\alpha) - \left((1-p_e) \cdot E\left[\tilde{f}_{n-\alpha}^{i,j}(\boldsymbol{w},\alpha) \right] + p_e \cdot E\left[\hat{f}_{n-1}^{i} \right] \right) \right| \tag{11}$$

$$CE_{(\boldsymbol{w},\alpha)} = \sum_{i=1}^{p} \left(\hat{f}_{n-\alpha}^{i,j}(\boldsymbol{w},\alpha) - f_n^i \right)^2 \tag{12}$$

Note that Eq.(11), following from Eq.(10), obviously accounts for the minimization of error propagation, while Eq.(12), related to the power of motion-compensated residuals, accounts for the bit rate required for encoding residuals. Since our purpose is to find MVs which can not only suppress possible error propagation, but also remove redundancies efficiently, a new constrained optimization problem based on $EP_{(\boldsymbol{w},\alpha)}$ and $CE_{(\boldsymbol{w},\alpha)}$ is thus formed below:

$$\underset{(\boldsymbol{w},\alpha)}{Min}\{EP_{(\boldsymbol{w},\alpha)}\} \qquad \text{subject to} : CE_{(\boldsymbol{w},\alpha)} \le \sigma^2 . \tag{13}$$

where σ^2 is a predefined power constraint. By controlling the threshold σ^2 , the magnitude of residual signals is restricted. This is also helpful to the validity in assuming a small $r_n^i(\Delta x, \Delta y)$ in deriving Eq.(10). Note that a small threshold σ^2 might cause no solution of Eq.(13), while a large σ^2 gives nearly no constraint on minimizing $EP_{(w,\alpha)}$. The solution of Eq.(13) can be easily found via a full search on possible $(\boldsymbol{w},\alpha)$'s. Note that our algorithm does not change the nature of full search in traditional ME process, but to provide a more proper criterion in selecting MVs that are expected to achieve both error resilience and code efficiency. Our criterion can also be applicable to fast ME where only a certain set of MV candidates are evaluated.

3 Experimental Results

The proposed algorithm was implemented on JM93 test model of H.264 video coding standard. The tested video sequences include “Akiyo”, “Table tennis”, and “Stefan” of CIF size. Each sequence contains 100 frames and is coded at 30 fps. Different quantization parameters (*QP*) (from 20 to 44) are applied to encode sequences for the evaluation of the proposed algorithm under a wide range of bit rates. Because our algorithm is to explore the error resilience of MVs, only the first frame is coded as I picture and the others are coded as P picture without intra-coded MBs therein. The block size for motion estimation is fixed to 16x16 pixels and three reference frames are allowed for each MB. Our proposed ERME algorithm is compared with the conventional search criterion based on SAD at two packet loss rates: 5% and 20%. It is assumed that each packet contains 22 successive MBs.

The experimental results are shown in Figs.2, 3 and 4. In these figures, the horizontal and vertical axes represent the average bit rate and the averaged PSNR of the reconstructed videos at the decoder, respectively. Notice that each erroneous MB at the decoder is concealed via the zero-motion algorithm, i.e., replaced by the MB at the same location of the previous frame. The curve marked with “PMV_X” represents our

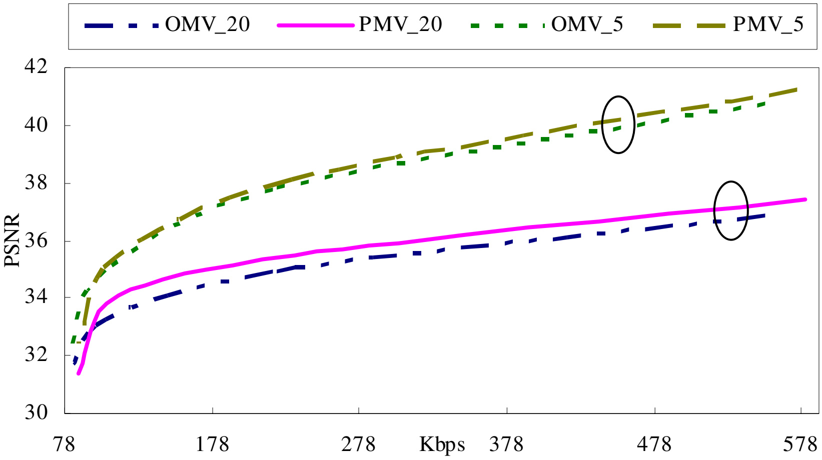


Fig. 2. Averaged PSNR curves of decoded “Akiyo” sequence with errors

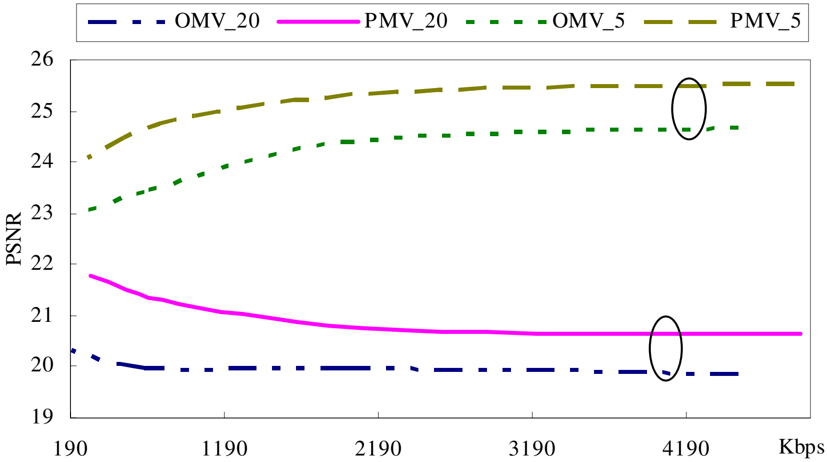


Fig. 3. Averaged PSNR curves of decoded “Stefan” sequence with errors

proposed algorithm and the other marked with “OMV_X” represents the original MVs obtained by full search algorithm with conventional SAD criterion; “X” represents the percentage of packet loss rate. For example, “PMV_5” represents the PSNR curve of the proposed algorithm at 5% packet loss rate.

It is observed from the experimental results that our proposed algorithm is approximate to the conventional ME algorithm in terms of the reconstruction PSNR for the video sequence “Akyio”. The reason is obvious: the inherited error propagation is minor due to the extremely static nature of “Akyio”. However, for video sequences with moderate to high motion, such as “Table tennis” and “Stefan”, in which the error propagation might be substantial, the effectiveness of our proposed algorithm in

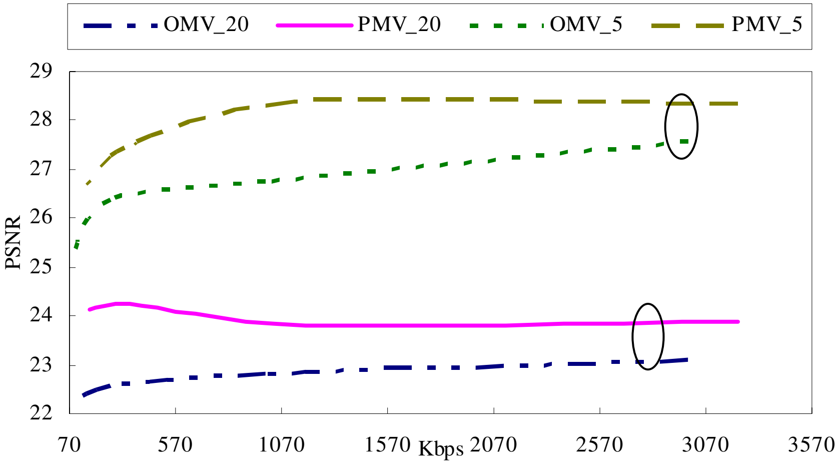


Fig. 4. Averaged PSNR curves of decoded "Table tennis" sequence with errors

suppressing error propagation becomes much more significant. For the "Stefan" sequence, our proposed algorithm can improve the PSNR performance by up to 1.0 dB at PLR = 20%. On the other hand, our algorithm might be inferior to conventional motion estimation algorithm at low bit rates and for static sequences. This is a result of making a tradeoff between coding efficiency and error resilience.

Finally, the choice of σ^2 may play an important role in controlling the effectiveness of our proposed algorithm. The tighter the constraint imposed by σ^2, the less effective the suppression of error propagation (since $EP_{(w,\alpha)}$ and $CE_{(w,\alpha)}$ might be contradictory for a given (w,α)). Currently, σ^2 is determined by summing the squared error for MB difference between frames and hence adaptive to MBs.

4 Conclusions

In this paper, a new error resilient ME algorithm is proposed. In the proposed algorithm, finding MVs becomes solving a constrained optimization problem in considerations of both coding efficiency and error propagation. From experimental results, it is observed that our proposed algorithm can improve the reconstruction quality by up to1.0 dB in presence of a PLR of 20% for videos with moderate to high motion. The contribution of our work is that an efficient and feasible algorithm, in contrast to others ([3], [4]) in literature, is proposed to optimally search MVs that are capable of suppressing error propagation and achieving high coding efficiency.

References

1. Yao Wang and Qin-Fan Zhu, "Error control and concealment for video communication: A review," *Proc. of IEEE*, vol. 86, no. 5, pp. 974 – 997, May 1998.
2. Yao Wang, Stephan Wenger, Jiangtao Wen, and Aggelos K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Processing Mag.***,** vol. 17, pp.61–82, July 2000.

3. Hua Yang, K. Rose, “Rate-Distortion optimized motion estimation for error resilient video coding,” *in Proc. of IEEE Int’l Conf. Acoustics, Speech, and Signal processing,* vol. 2, pp. 173-176, March 2005.
4. Oztan Harmanci and A. Murat Tekal, “Stochastic frame buffers for rate distortion optimized loss resilient video communications,” *in Proc. of IEEE Int’l Conf. Image Processing,* vol. 1, pp. 789-792, Sept. 2005.
5. Zhihai He, Jianfei Cai and Chang Wen Chen, “Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding,” *IEEE Trans on CSVT*, vol 12, no 6, pp. 511-523, June 2002.
6. R. Zhan, S. L. Regunathan, and K. Rose, “Video coding with optimal intra/inter mode switching for packet loss resilience,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 6 pp. 966-976, June 2000.
7. Jorn ostermann, Jan Bormans, Peter List, Detlev Marpe, Matthias Narroschke, Fernando Pereira, Thomas Stockhammer, and Thomas Wedi, “Video coding with H.264/AVC: tools, performance, and complexity,” *IEEE Circuit and Systems Mag.*, vol. 4, pp. 7-28, First quarter 2004.
8. Pao-Chi Chang, Tien-Hsu Lee, Jhin-Bin Chen, and Ming-Kuang Tsai, “Encoder-originated error resilient schemes for H.264 video coding,” *18th IPPR Conference on Computer Vision, Graphics and Image Processing,* pp. 406–412, Aug. 2005.
9. A. Leontaris and P.C. Cosman, “Video compression for lossy packet networks with mode switching and a dual-frame buffer,” *IEEE Trans. on Image Processing*, vol. 13, pp. 885-897, July 2004.
10. G. Cote and F. Kossentini, “Optimal intra coding of blocks for robust video communication over the Internet,” *Signal Processing: Image Commu.*, pp. 25-34, Sep.1999.

A Novel Scrambling Scheme for Digital Video Encryption

Zhenyong Chen[1], Zhang Xiong[1], and Long Tang[2]

[1] School of Computer Science and Engineering, Beihang University,
Beijing, China, 100083
[2] Department of Computer Science and Technology, Tsinghua University,
Beijing, China, 100084
chzhyong@buaa.edu.cn, xiongz@buaa.edu.cn,
tanglong@mail.tsinghua.edu.cn

Abstract. Nowadays it is easy and fast to deliver and communicate the digital video contents. Therefore, security problem reveals followed in applications such as video-on-demand, videoconference and video chatting system. In this paper, a scrambling scheme employing novel grouping strategy is proposed to encrypt the compressed video data. In new scheme, the scrambling and descrambling operations are embedded into the video encoding and decoding processes. DCT coefficients are divided into 64 groups according to their positions in 8×8 size blocks, and scrambled inside each group. Besides, motion vectors are also grouped and permuted in term of their modes. Thus the influence on statistical property of the video data is reduced furthest. Experimental results show that the proposed method has less bitstream overhead and no any image quality degradation. It is also indicated that the security is enhanced considerably via extending the scrambling range. A content-based dynamic secret key method is also presented to withstand the known-plaintext attack.

Keywords: Video Encryption; MPEG-4; Scrambling.

1 Introduction

Distributed multimedia applications such as video on demand system, video broadcast, video conferencing and videophone make the multimedia security research a key issue [1]. Intended tortious behavior will even cause an unprotected video communication leak the privacy of communicators. Thus content creators and providers will be reluctant to communicate or distribute their video contents if they are not assured that contents are securely protected. In this instance, methods of video encryption with conditional access act as one key technique to secure the video content.

In the past years, methods to encrypt video data have been discussed can be classified into three categories, that is, cryptographic method, permutation method and hybrid or other methods. A typical cryptographic approach called naive algorithm is to encrypt the entire MPEG stream using standard encryption algorithms like DES

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 997 – 1006, 2006.

(Data Encryption Standard) [2] and AES (Advanced Encryption Standard). Naive algorithm treats the MPEG bit-stream as ordinary data and does not use any of the special MPEG structure. Although this is supposed to be the most secure MPEG encryption algorithm that preserves the data size, due to high data rate of video, it usually adds a large amount of processing overhead. There are some other cryptographic approaches called selective algorithms [3, 4] that use the features of MPEG layered structures. These methods partially choose the portions of the compressed video such as headers, I frames and I-blocks in P and B frames, and encrypt them using standard encryption algorithms.

Tang [5] first presents a zig-zag permutation method for MPEG video. The encryption operation is embedded into the MPEG compression process. Instead of mapping the 8×8 block to a 1×64 vector in zig-zag order, it uses a random permutation list to map the individual 8×8 block to a 1×64 vector. Zeng [6] extends the permutation range to the segment and each segment consists of several macroblocks/blocks. Within each segment, DCT (Discrete cosine transform) coefficients of the same frequency band are randomly shuffled within the band. Besides, there is another permutation method that shuffles the code words in the VLC (Variable length coding) tables [7].

Qiao and Nahrstedt present a special method named Video Encryption Algorithm (VEA) to encrypt the video data [8]. It utilizes the results of the statistical analysis of compressed video frames, encodes half of the data using DES, and has a 47% gain in terms of number of XOR operations over DES. VEA provides efficient, real-time and secure encryption for low-resolution and low bit rate MPEG sequence. However, it will have real-time implementation problem for high resolution and high bitrate sequences.

In this paper, we present a new scrambling scheme to encrypt the video data via grouping the DCT coefficients and motion vectors, which has less bitstream overhead and higher security than prior similar encryption schemes and has no any image quality degradation. Furthermore, a content-based dynamic generating method of secret keys is given to increase the ability resisting to the known-plaintext attack. The remainders of the paper are structured as follows. Section 2 discusses the group-based scrambling scheme for DCT coefficients in frequency domain. Section 3 gives a scheme to shuffle the motion vectors of MPEG-4 video. Then we perform experiments and security analysis in section 4. Finally, in section 5, we draw some conclusions.

2 Frequency Domain Permutation

It is a natural idea to distort the visual image directly in the spatial domain. Before comeback the distorted video appears unintelligible and this method is employed in several video scrambling systems [9, 10]. But it will significantly change the statistical property of the original video signal, thus making the original video very difficult to compress. Another potential drawback of scrambling video in the spatial domain is that the highly spatially- and temporally-correlated nature of the video data can be used for efficient attacks [6]. So we address the permutation method in the frequency domain.

All current international standards for video compression, namely MPEG-1, MPEG-2, ITU-T H.261, ITU-T H.263, and the baseline mode of MPEG-4 and H.264 are hybrid coding schemes. Such schemes are based on the principles of motion compensated prediction and block-based transform coding. The transform used is often the discrete cosine transform. Fig. 1 shows a generic block diagram of a hybrid coding scheme. The video sequence to be compressed is segmented into groups of pictures (GOP). Each GOP has intra-coded frame (I-frame) followed by some forward predictive coded frames (P-frames) and bidirectional predictive coded frames (B-frames). I frames are split into nonoverlapping blocks (intra-coded blocks) of 8×8 pixels which are compressed using DCT, quantization (Q), zig-zag-scan, run-level-coding and entropy coding (VLC). P/B frames are subject to motion compensation by subtracting a motion compensated prediction. The residual prediction error signal frames are also split into nonoverlapping blocks (inter-coded blocks) of 8×8 pixels which are compressed in the same way as blocks of inter-frames. Sometimes, P/B frames also have some intra-coded blocks when better efficiency will be obtained using intra-coded compression. For one video standard, when setting a certain group parameters, the bitrate and visual quality of the compressed video data will be fixed always.

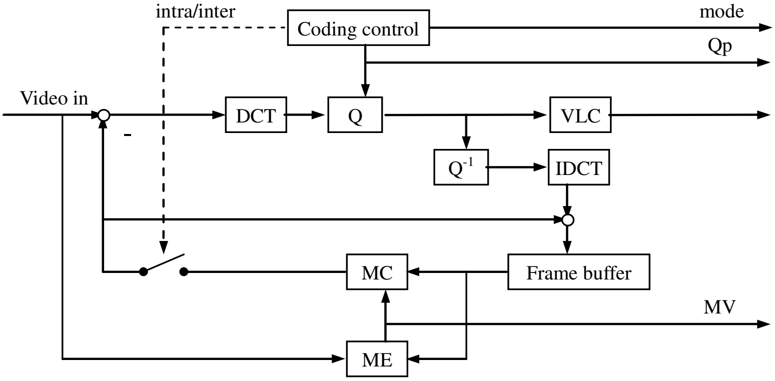


Fig. 1. Hybrid video coding scheme

In order to avoid the visual quality degradation, we shuffle DCT coefficients after the quantization stage. In other words, the scrambling module is embedded into the position between the quantization and VLC stages. In our scheme shown in Fig.2, the scrambling module contains three components, the secret key K_D, the shuffling table generator STG, and the scrambler S. Using the secret key, STG generates a shuffling table for the quantized DCT coefficients. Then the scrambler permutes the quantized DCT coefficients according to the shuffling table and the remaining procedures are the same as the generic hybrid coding scheme. A corresponding reverse process to recover the quantized DCT coefficients is given in Fig.3.

In 8×8 DCT transform coding, the 64 transformed coefficients are zig-zag ordered such that coefficients are arranged approximately in the order of increasing frequency. This arrangement enhances the efficiency of the run length coding because the bitstream coder uses a run-length and variable-length coding technique that generally assigns shorter codewords to combinations of coefficient values and run lengths. So breaking the zig-zag order within the block will significantly degrade the statistics relied upon by a run-length coder. So we use a new grouping strategy that is illustrated in Fig.4 to shuffle the DCT coefficients. In fact, the DCT coefficients in the same frequency position have most approximate statistics property. According to this point, for each signal, all the DCT coefficients in a frame are partitioned into 64 groups by the frequency position and then shuffled inside each group. In Fig.4, *n* is total number of 8×8 blocks of each signal, for luminance signal, *n*=*w*×*h*/8/8, and for chrominance signals, *n*=*w*×*h*/16/16, where *w* is the width of frames and *h* is the height of frames. In some extent, except for the optimization method, this strategy can be regarded as the method that has the least influence on the statistics property of the quantized DCT coefficients. Thus the overhead of the compressed bit-stream will be considerably limited.

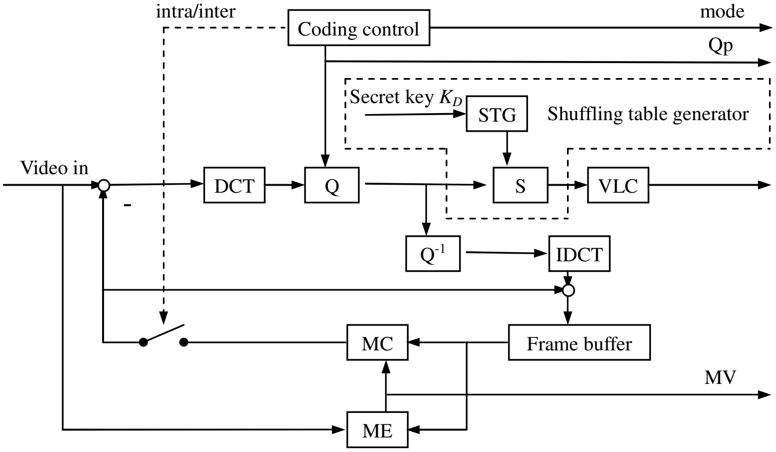


Fig. 2. Scrambling of quantized DCT coefficients

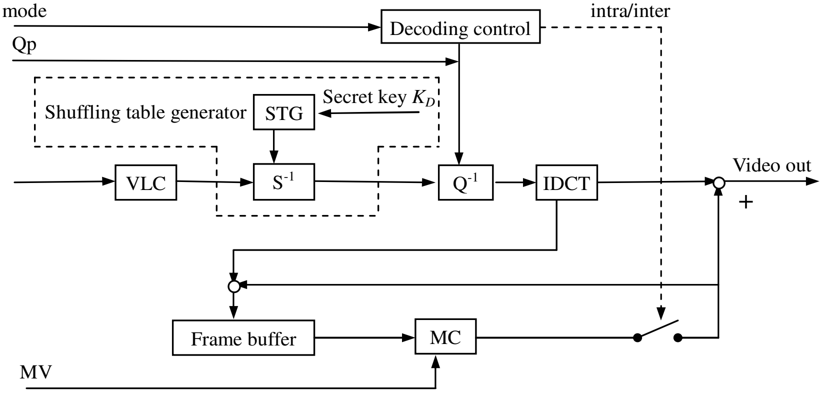


Fig. 3. Recovery of scrambled quantized DCT coefficients

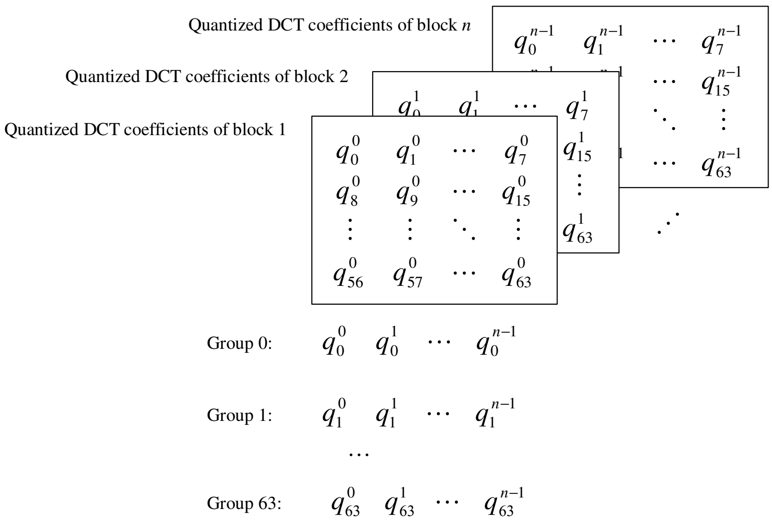

Quantized DCT coefficients of block n
Quantized DCT coefficients of block 2
Quantized DCT coefficients of block 1
q_0^{n-1} q_1^{n-1} $\cdots$ q_7^{n-1}
q_{15}^{n-1}
q_{63}^{n-1}
q_0^1 q_1^1 $\cdots$ q_7^1
q_{15}^1
q_{63}^1
q_0^0 q_1^0 $\cdots$ q_7^0
q_8^0 q_9^0 $\cdots$ q_{15}^0
q_{56}^0 q_{57}^0 $\cdots$ q_{63}^0
Group 0: q_0^0 q_0^1 $\cdots$ q_0^{n-1}
Group 1: q_1^0 q_1^1 $\cdots$ q_1^{n-1}
$\cdots$
Group 63: q_{63}^0 q_{63}^1 $\cdots$ q_{63}^{n-1}

Fig. 4. Grouping of quantized DCT coefficients

When there is less motive objects and scene, the scrambling of I frames will render the following P/B frames difficult to perceive due to the dependency of P/B frames on I frames. However, in many other circumstances, the video content may not be highly correlation in temporal dimension. In these cases, the intra-coded blocks and the accumulating residual energy will leak the partial image information. In order to achieve higher security, the DCT coefficients of intra-coded blocks and non intra-coded blocks in P/B frames need to be shuffled also. Apparently, when employing same grouping strategy, they should be separated each other.

3 Motion Vector Permutation

Theoretically, it is absolutely secure in visual perceptivity when I frame, intra-coded blocks and inter-coded blocks in P/B frames are all encrypted but the bit-stream overhead will reach a certain level. For inter-coded blocks in P/B frames, there is an optional substitutive scheme that encrypts their motion vectors because both the similar encryption efficiency and less bit-stream overhead will be obtained. Additionally, in some scenarios, it is necessary to scramble the motion vector information for those applications that have high level security requirement. In Fig.5 similarly to the scrambling scheme of DCT coefficients, it can be seen that there are three components included in the motion vector encryption module, the secret key K_M, the shuffling table generator STG, and the scrambler S. Besides, a reverse process to recover the scrambled motion vectors can be seen in Fig.6.

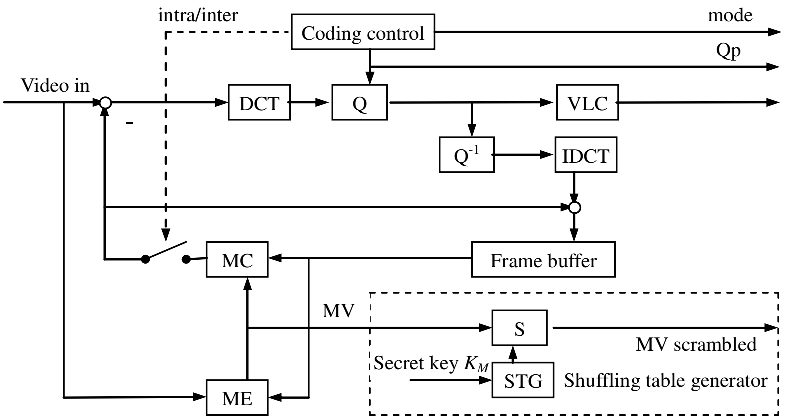


Fig. 5. Scrambling of motion vectors

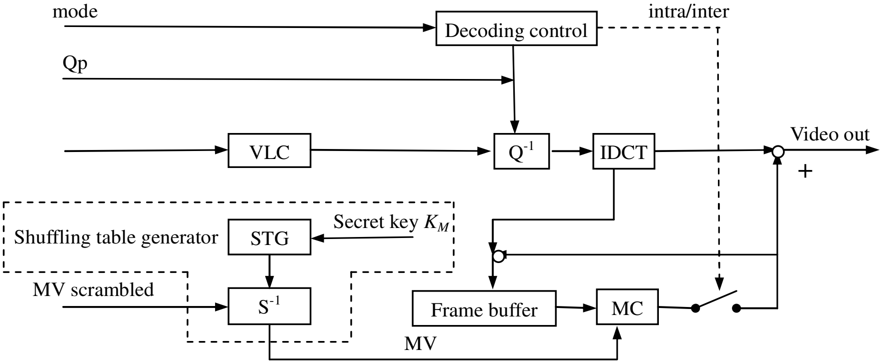


Fig. 6. Recovery of scrambled motion vectors

Generally in hybrid video coding scheme, the motion estimation and compensation are carried out based on the choice of 16×16 blocks (referred to as macroblocks) as a basic unit. The estimated motion vectors and other side information are encoded with the compressed prediction error in each macroblock depending on the type of the macroblock. For example, in MPEG-4 standard, the macroblock of P frame has three type motion vectors listed in Table 1, that is, 16×16, 16×8 and 8×8. Different type macroblock has different number motion vectors and they are differenced with respect to a prediction value and coded using variable length codes. So we group motion vectors in terms of their types in the same way as we group the quantized DCT coefficients and then shuffle them inside each group. Thus the encrypted motion vectors can be recovered in the decoder side correctly. In addition, B frame has more type motion vectors but the same strategy can be used.

Table 1. Types of motion vectors of P frame

Types	Numbers of motion vector(s)	Has4MVForward Flag	FieldMV Flag
16×16	1	FALSE	FALSE
16×8	2	FALSE	TRUE
8×8	4	TRUE	Ignored

4 Experiments and Security Analysis

4.1 Experiments

In order to evaluate the performance of the new proposed schemes, we implement a demo via integrating our encryption strategies into the MPEG-4 verification model coder [11] provided by ISO. Using this demo, experiments to scramble I frame, P frame motion vectors respectively and both of them are performed. The test video sequences with QCIF (176×144) size include Foreman, CarPhone, Claire, Grandma, MissAm, Salesman, Trevor and Suzie, which are shown in Fig.7.

a) Foreman b) MissAm c) CarPhone d) Grandma

e) Salesman f) Claire g) Travor h) Suzie

Fig. 7. Original video sequences

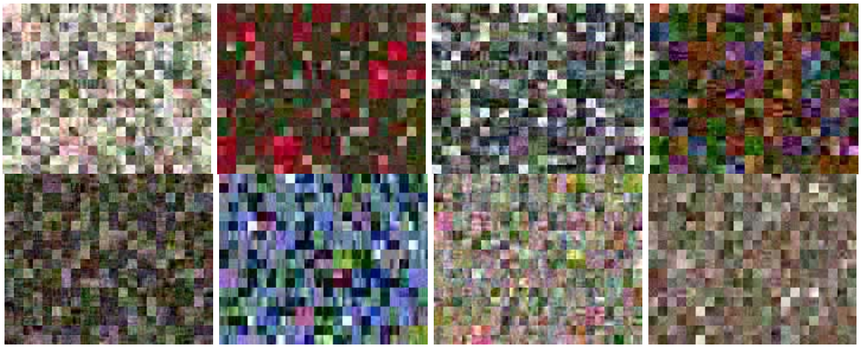


Fig. 8. Scrambling of quantized DCT coefficients of I frame

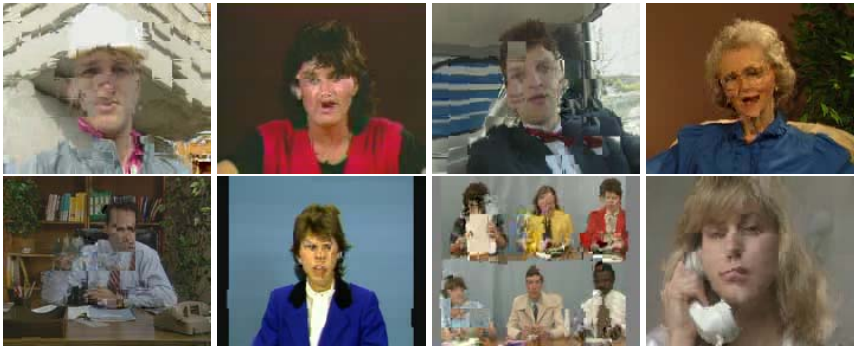


Fig. 9. Scrambling of P frame motion vectors

Fig.8 gives a scrambled I frame image of each test video sequence and the details of images are not visible completely. Fig.9 shows that macroblocks of P frame existing motion, that is, inter-blocks, have been thrown into confusion, which benefits to keep the secret of the motion information. In Table 2, it can be seen that the bitstream overhead of I frame scrambling is in the range from 9.54% to 48.16% and the Foreman video is lowest. It seems that the motive videos have less bitrate increase than those still videos relatively. The same results of P frame motion vector scrambling can be found also. However, like MissAm, Grandma and Claire, the videos that have high relative bitstream overhead still have rather high compression rate. This indicates that our encryption schemes are acceptable in the manner of influence on compression efficiency. Zeng [6] gave the bitstream overhead of Carphone video by means of his own method. So we use the experimental data of Carphone video to compare with his result, which are listed in Table3. It shows that our method is better.

Table 2. Experimental results

Video Sequences	Total number of frames	Original length	Normally compressed length	I frame permutation Overhead	I frame and P frame motion permutation	
					Overhead	Compression rate
Foreman	400	15,206,400	335,529	9.54%	11.27%	40.73
MissAm	150	5,702,400	36,642	28.19%	67.03%	93.17
Carphone	382	14,522,112	302,281	11.73%	16.56%	41.22
Grandma	870	33,073,920	264,765	23.47%	41.74%	88.13
Salesman	449	17,069,184	192,104	11.37%	17.04%	75.92
Claire	494	18,779,904	114,272	48.16%	84.34%	89.15
Travor	150	5,702,400	90,583	12.26%	17.83%	53.43
Suzie	150	5,702,400	66,490	13.53%	23.20%	69.61

Table 3. Comparison of bit overhead of Carphone video sequence

Scrambling mode	Zeng's method	Our method
Quantized DCT coefficients of I frames scrambled	19.8%	11.7%
Both quantized DCT coefficients of I frames and motion vectors of P frames scrambled	23.6%	16.6%

4.2 Security Analysis

For a 352×288 size I frame, assuming that $n_{\mathrm{Y}i}$ nonzero coefficients of Y luminance signal, $n_{\mathrm{U}j}$ nonzero coefficients of U chrominance signal and $n_{\mathrm{V}k}$ nonzero coefficients of V chrominance signal exist in the position respectively indexed by i, j and k, the times of required trials in which an attacker attempts to recover the original I frame is

$$\prod_{i=1}^{64} n_{Yi}! \times \prod_{j=1}^{64} n_{Uj}! \times \prod_{k=1}^{64} n_{Vk}!$$

where $n_{Yi} \le 44 \times 36 = 1584, n_{Uj} \le 22 \times 18 = 396, n_{Vk} \le 22 \times 18 = 396.$ It can be seen that the computation cost is rather tremendous to completely recover the original I frame. Similarly, if the attacker wants to recover the motion vectors scrambling of P frame, the number of required trials is $r! \times s! \times t!$, where r is the number of 16×16 type motion vectors, s is the number of 16×8 type motion vectors and t is the number of 8×8 type motion vectors.

To strengthen the ability withstanding the known-plaintext attack, the content-based dynamic secret key generating method can be employed. For example, the encryption key of the quantized DCT coefficients can be produced as follows.

$$K_D(p) = KF(K_m, F_p), p = 1,2,\cdots,P \tag{1}$$

where p is the index of frames, $K_D(p)$ is the secret key of p frame, KF the key generation function, K_m the main key to generate dynamic keys, Fp the feature of p frame, and P is the total number of frames. The feature can be extracted from the content characteristics of each frame, like sum or average of the quantized DCT coefficients which should not change after shuffling operation. Thus each frame has individual scrambling table and this will make the known-plaintext attack more difficult. Additionally, the key synchronization also can be held correctly and will not be destroyed by the frame desynchronization such as incidental frame dropping and adding.

5 Conclusion

In order to enhance the security, it should be regarded as one principle that the encryption must scramble the image and motion information in a best possible extent because illegal attackers will impose this information possibly. Moreover, shuffling operations modify statistical property of the video data in a certain degree and this will cause the increase of the compressed video bitstream. Therefore, it is also necessary to be treated as another principle that the scrambling operation must bring possibly less influence on the statistical property of the video data. According to these two principles, we present a new scrambling scheme to encrypt the video data via employing the new grouping strategy. Experimental results show that the novel scrambling scheme has less influence on the compressed video bitstream. In addition, a content-based dynamic generating method of secret keys is given to increase the ability resisting to the known-plaintext attack. The analysis also indicates that the new scheme has rather high security. It needs to be emphasized that the encrypted video data has standard video format. So using this method, we can open the beginning part or any segments of a movie to attract the audience and keep secret of the remainder.

Acknowledgments. This work is supported by a grant of key program named "Theory and Methods of Security for Image/Graphics Information on Internet (No.60133020)" from National Natural Science Foundation of China. The demo used in our experiment is implemented based on the MPEG-4 reference source code, which is provided by International Organization for Standardization. In addition, using the open source of XVID MPEG-4 codec, we also implement a video encryption/ decryption plug-in component for common video players. So we would like to thank them together.

References

1. Lintian Qiao and Klara Nahrstedt. Comparison of MPEG encryption algorithms. International Journal of Computers and Graphics, special issue: "Data Security in Image Communication and Network" vol.22 January 1998.
2. I. Agi and L. Gong. An empirical study of secure MPEG video transmissions [A]. The Internet Society Symposium on Network and Distirbuted System Security [C]. San Diego, CA, 1996, 137-144
3. Y. Li, Z. Chen, S. Tan, et al. Security enhanced mpeg player. In proceedings of IEEE First International Workshop on Multimedia Software Development (MMSD'96), Berlin, Germany, March 1996.
4. T. B. Maples and G. A. Spanos. Performance study of a selective encryption scheme for the security of networked, Real-Time video. In Proceedings of 4th International Conference on Computer Communications and Networks, Las Vegas, Nevada, September 1995.
5. L. Tang. Methods for encrypting and decrypting MPEG video data efficiently [A]. Proceedings of the 4th ACM International Multimedia Conference [C]. Boston, MA, 1996, 219-229
6. Wenjun Zeng and Shawmin Lei. Efficient frequency domain selective scrambling of digital video [J]. IEEE Transactions on Multimedia, 2003, 5(1): 118-129
7. Mohan S. Kankanhalli and Teo Tian Guan. Compressed domain scrambler/descrambler for digital video. IEEE Transactions on Consumer Electronics, Vol.48, No.2, pp.356-365, May 2002.
8. Lintian Qiao and Klara Nahrstedt. A new algorithm for MPEG video encryption. In Proceedings of the First International Conference on Imaging Science, Systems and Technology (CISST'97), Las Vegas, Nevada, July 1997, pp.21-29.
9. G. L. Hobbs, Video scrambling, U.S. Patent 5 815 572, Sept.29, 1998.
10. D. Zeidler and J.Griffin, Method and apparatus for television signal scrambling using blocl shuffling. U.S. Patent 5 321 748, June 14, 1994.
11. ISO. Source code of MPEG4 encoder and decoder of publicly available standards [EB/OL]. http://www.iso.org/iso/en/ittf/PubliclyAvailableStandards/14496-5_Compressed_directories/ Visual/Natural.zip, 2004-03-20

An Electronic Watermarking Technique for Digital Holograms in a DWT Domain

Hyun-Jun Choi[1], Young-Ho Seo[2], Ji-Sang Yoo[1], and Dong-Wook Kim[1]

[1] Kwangwoon University, Welgye-Dong, Nowon-Gu, Seoul 139-701, Korea
[2] Hansung University, Samsung-Dong 3ga, Sungbuk-Gu, Seoul 136-792, Korea

Abstract. Digital hologram generated by a computer calculation is one of the most expensive contents and its usage is being expanded. Thus, it is highly necessary to protect the ownership of digital hologram. In this paper a spatial-domain and a frequency-domain electronic watermarking schemes were proposed. The spatial-domain scheme was only to compare the results to the ones from frequency-domain scheme and the frequency-domain scheme used 2-dimensional mallat-tree discrete wavelet transform. Both of them showed very high imperceptibility and quite high robustness against the attacks. Especially the MDWT-domain scheme was very high robustness such that the error ratio at the worst case was only 3%. Thus, we expect that it is used as a good watermarking scheme of digital hologram with high performance. But the spatial-domain scheme was turned out to be useless when data compression process is necessary after watermarking.

1 Introduction

As the digital era has been advanced since late 20th century, digital data has been getting more portions in data usage. This is because digital data has the advantages over analog data in easiness in storage, copy, adjustment, high immunity to noise in communication, etc. But some of these advantages act as its disadvantages: easy to illegal duplication, modification, etc. and not easy to distinguish the original from the duplicated. Accordingly, various digital contents are increasingly duplicated and used illegally by many internet users at present. Therefore, an issue of such illegal duplication/modification of digital contents such as protecting the ownership of intellectual properties is emerging.

Digital watermarking technology implements preparation of a data inside the content for claiming such ownership to protect the intellectual property right and it has been known as the best solution for this protection. So far, many researchers have been studying this technology for many valuable digital contents such as audios, images, moving pictures, etc. Digital watermarking is a technique that a specified data (watermark) is concealed in a digital content but it is almost impossible to know whether the watermark is embedded or not (imperceptibility) when it is actually used. Its another very important property is that the embedded watermark should not be harmed or removed by malicious or

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1007–1017, 2006.

non-malicious attacks (robustness) so that the extracted watermark after attack should be until the original content is useful [1].

Digital hologram is a technique that the interference patterns between the reference light wave and the object light wave is captured with a CCD camera instead writing it on a holographic film [2]. The original image can be reconstructed by loading the digital hologram on a Spatial Light Modulator and illuminating the reference light that is the same as was recording. Other method to obtain a digital hologram is by calculating the interference patterns on a computer (Computer-Generated Hologram, CGH). A CGH is very time-consuming because each pixel value is affected by each source point of the object. Thus a hologram or a digital hologram is relatively very expensive 3-dimensional image that recently, researchers in many institutes in the world are studying hologram and its watermarking related technology. But most of them are the optical methods with optical elements or optical parameters [3,4].

In this paper, we propose a digital watermarking method for digital holograms electronically, not optically. The main idea is to use frequency-domain characteristics of digital holograms, for which DWT (Discrete Wavelet Transform) is used to transform into a frequency-domain. Because all the existing watermarking methods are optical ones, we set up a spatial-domain electronic watermarking method for the comparison purpose. In this paper, all the holograms are assumed digital, that is, fringe patterns captured by CCD camera or generated by a computer.

2 Analysis of Digital Holograms

In this section, digital holograms, that is, fringe patterns are analyzed to determine the proper locations to insert digital watermark in. From now on, a fringe pattern is regarded as a spatial data, while the frequency-domain data is the results from transforming the fringe pattern by a frequency-domain transform technique, that is, DWT. Also, we assume that each pixel in a fringe pattern has a grey-level value. It can be assumed that a fringe pattern is colored. In this case also, the color components are separated into red, green, and blue images. The image in this paper corresponds to one of the three components.

Fig. 1 (a) shows the hologram image (200×200 [$Pixels^2$]) created by computer graphics and 1 (b) is its fringe pattern ($1,024 \times 1,024$ [$Pixel^2$]), that is, CGH created by means of numerical computation with the image. In fact, a fringe pattern itself is a frequency-domain data transformed by Fresnel transform. But as mentioned before, this paper considers a fringe pattern as a spatial data.

Fig. 2 shows the procedure to analyze the fringe patterns in the spatial domain and the frequency domain. In both analysis procedures, a fringe pattern is divided into several segments. From the properties of a fringe pattern, a part of a fringe pattern can reconstruct the original image with some degradation in image size according to the size of the segment. That means a data in a segment affects reconstruction of the whole image. Thus, a part of fringe pattern can be used for watermarking without loosing the imperceptibility and robustness of watermark.

For spatial analysis, each bit-plane (BP) of the fringe pattern is examined as expressed in Fig. 2 (a). In this analysis the strength of data that affects the quality of the reconstructed image seriously is examined. For frequency-domain analysis, 2-dimensional (2D) transformed are performed by regarding a fringe pattern as a 2D data. Both mallat-tree decomposition (MDWT) method and packetization DWT (PDWT) method are considered. 2D MDWT is the standard transform in JPEG2000.

The watermark embedding process might disturb the energy distribution of a target data because embedding watermark changes the values of data. The change in energy distribution in turn affects some characteristics of the data, which might change the results of further processes such as signal processing or data compression. Therefore, the analysis in this paper mainly consists of the processes to examine the energy distribution of the data with considering the watermarking process. In this paper, our watermarking scheme is to replace a specific bit of a specific position with a digital watermark bit.

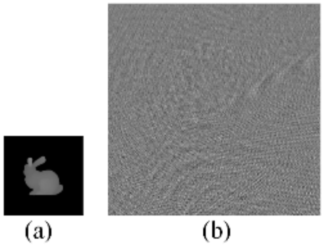


Fig. 1. (a) Original image (b) fringe pattern of (a)

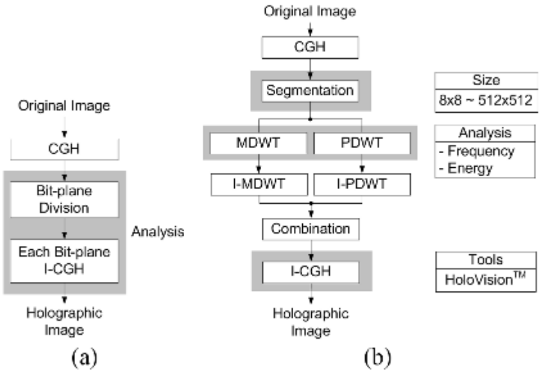


Fig. 2. Procedure for analysis: (a) spatial-domain (b) frequency-domain

2.1 Analysis of a Fringe Pattern in Spatial-Domain

One of the candidate data domains for watermarking might be the spatial domain. As mentioned before, the fringe pattern itself is regarded as the spatial data in this paper. Also, we consider the whole fringe pattern or one of the segmented one as the target watermarking space. The feature analysis of a fringe pattern in the spatial domain consists of the examining the effect of watermarking on a specific bit-plane (BP). Here in this paper, a pixel of in a fringe pattern consists of 8-bits for its value as shown in Fig. 3.

The main purpose for spatial-domain data analysis is how much a specific BP affects to the reconstructed image (result from inverse CGH (I-CGH)) with a fringe pattern. Thus, we examine the reconstructed result for each BP of a fringe pattern and the example images are shown in Fig. 4. In real experiment, more than 100 fringe patterns were examined.

As expected from the knowledge from 2D images, BP7 (most significant BP, MSBP) alone reconstructs the image with a little degradation in the intensity. This is because of the weight of the BP7 is almost half of the value. An unexpected result from the experiment is that the BP6 of the most fringe patterns can reconstruct the image recognizably, even though the intensities are pretty much degraded. It means that only 1/4 in intensity of a fringe pattern can reconstruct the image, which might be usefully used in data compression of a digital hologram. In most of the fringe patterns, a BP lower than BP6 cannot reconstruct the image, as shown in Fig. 4 (d). It means that data adjustment in BP lower

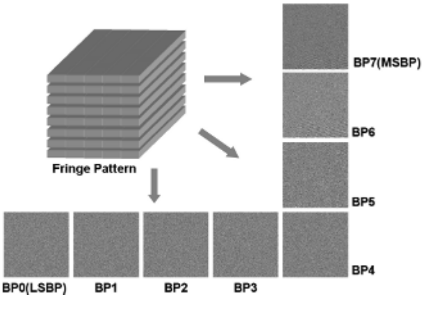


Fig. 3. Bit-planes of a fringe pattern or its segment

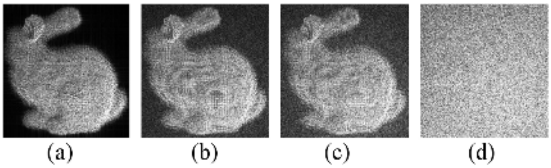


Fig. 4. Reconstructed images by HoloVision: (a) original (b) with BP7 (c) with BP6 (d) with BP5

than BP6 hardly affect the reconstructed image and they might be good places for watermarking.

2.2 Analysis of a Fringe Pattern in Frequency-Domain by DWT

Frequency analysis by 2D DWT is to examine the energy distribution throughout the frequency bands because the energy distribution is the most important to decide the watermarking positions with maintaining the imperceptibility and robustness of the embedded watermark.

Mallat-tree 2D DWT. MDWT is to decompose an image into monotonically increasing frequency subbands. In a level of transform, an image in decomposed into four subband: *LL, HL, LH,* and *HH,* where *'L' ('H')* means low-pass filtered result and the first (second) letter of the two is for horizontal (vertical) direction. For example, *HL* means the resulting subband from high-pass filtering in horizontal direction and low-pass filtering in vertical direction. Once a level of transform is completed, the next level of transform is for the lowest subband, *LL*. Thus, k-level decomposition produces *3k+1* subbands. Fig. 5 shows an 5-level 2D MDWT results where (a) shows the decomposition scheme and the subband numbers and (b) is an example with Lena 2D image.

The average energy distribution of the result from 5-level 2D MDWT for 100 fringe pattern is shown in Table 1, in which each energy value is the total energy of the coefficients in the corresponding subband. That means the energy value of each coefficient in a subband is 4 times higher that the one in the table than the one in the one-level lower subband because the number of coefficients in one-level lower subband is 4 times larger.

In general, a fringe pattern is known to have much high-frequency components compared to an ordinary 2D image. But from the experiment, a fringe pattern showed a similar property of very high energy concentration (more than 70 % on the lowest frequency components, that is, subband 0 in this experiment.

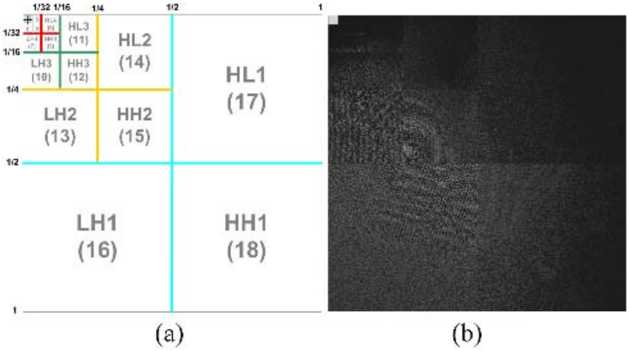


Fig. 5. Frequency domain of 5-level MDWT: (a) scheme and subband numbers (b) transformed example

One peculiar characteristic in this experimental result is that some of the higher frequency components (subband 10, 12, 13, 15, 16, 18) have quite high energy distribution. These subbands are in the vertical or diagonal direction which means a fringe pattern has quite high energy in vertical and diagonal-directional high components. Consequently, the lowest, the vertical, and the diagonal subbands are good candidates for watermarking place.

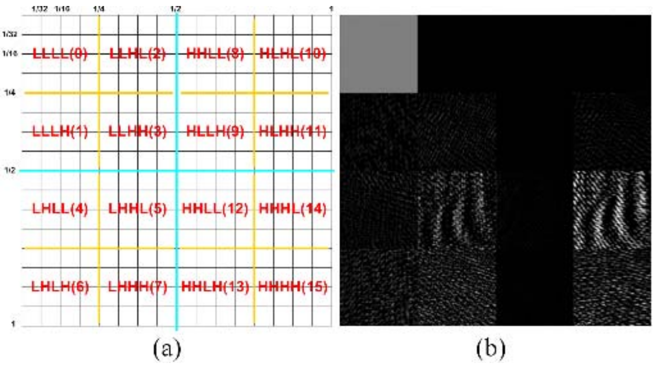
Packetization 2D DWT. One modified 2D DWT method is packetization 2D DWT (2D PDWT). The scheme is as follows. The first level of transform is the same as MDWT. But from the second level, the transform proceeds with a particular purpose. For example, if local energy concentration is needed, a subband whose energy is lower than a given threshold value is continuously transformed until the resulting energy of subband is higher than the threshold. In Fig. 6, all the subbands except *HH* proceeded the second-level of transform. Among the subbands from *LL*, only *HL, LH* subband was transformed in the third level and only *LL, HL, LH* subband was transformed in the fourth level, and so on.

Table 1. Average energy distribution after 5-level 2D MDWT

Subband	*Average energy*	*Ratio*	*Subband*	*Average energy*	*Ratio*
0	16199.634	72.677	10	2.208	0.010
1	0.006	0.000	11	0.883	0.004
2	0.022	0.000	12	12.514	0.056
3	0.077	0.000	13	224.675	1.008
4	0.051	0.000	14	6.705	0.030
5	0.024	0.000	15	567.815	2.547
6	0.104	0.000	16	2185.485	9.805
7	0.214	0.001	17	19.361	0.087
8	0.283	0.001	18	3068.226	13.765
9	1.559	0.007	Total	22289.845	100.000

However, this scheme is not easy to apply for examination of digital holograms because each one may have different frequency characteristics. Thus in this paper, we generalize 2D PDWT as Fig. 7. Here, we only transformed two level of 2D DWT and each level-1 subband was transformed for one more level with any restriction in proceeding further transform. Thus all the subbands are the results of the two-level 2D DWT.

The average energy distribution is shown in Table 2. Differently from MDWT, the results from PDWT showed quite random distribution in energy. Subband 0 has the highest energy but subband 14 also has very high energy. Also, subband 5, 6, 7, 15 retain quite high energy. Thus, PDWT does not look good transformation to find the watermarking locations with the energy distribution. Consequently we use only MDWT to find the watermark positions.



(a)

(b)

Fig. 6. Frequency domain of 2D PDWT: (a) scheme and subband numbers (b) transformed example

Table 2. Energy distribution and ratio of fringe pattern

Subband	*Average energy*	*Ratio*	*Subband*	*Average energy*	*Ratio*
0	16201.034	38.357	8	0.268	0.001
1	224.675	0.532	9	5.734	0.014
2	6.705	0.016	10	13.869	0.033
3	567.815	1.344	11	274.797	0.651
4	625.367	1.481	12	133.677	0.316
5	1533.207	3.630	13	54.296	0.129
6	3743.395	8.863	14	10972.849	25.979
7	3813.564	9.029	15	4066.018	9.627
	Total			42237.271	100.000

3 Electronic Watermarking Method on a Digital Holograms

In this section, we propose an electronic watermarking method for digital hologram. To find the watermarking positions and embed the watermark, we use 2D MDWT. But we also propose a watermarking method on the spatial domain for the purpose of comparison with the frequency-domain method. In watermarking, we assume that the watermark data to be embedded consists of binary data. Especially we use a binary image with the size of 32 × 32 $[Pixels^2]$. This assumption is not a special one because the target data to be watermarked is also a digital and if a non-binary or analog data is to be used, it can be changed into a binary bit stream.

3.1 Watermarking in Spatial-Domain

In general, watermarking in a spatial domain has been done to the regions with large high-frequency components because of the imperceptibility of the watermark. Different from a general 2D image, a fringe pattern retains high-frequency

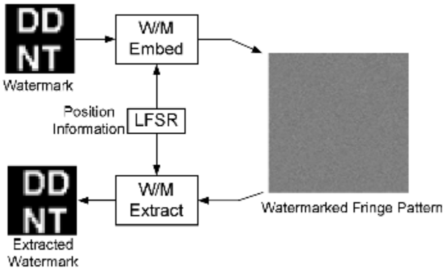


Fig. 7. Watermarking procedure in the spatial domain

components all around the image, which is experimentally examined in the previous section. Thus, electronic watermarking on a fringe pattern can be done any region of a fringe pattern. In this paper, we embedded the watermark data randomly to all around the fringe pattern, as shown in Fig. 7. To select the watermarking places, we used a 32-bit linear feedback shift register (LFSR) whose feedback characteristic is as follows.

$$P(x) = x^{32} + x^{22} + x^2 + 1 \tag{1}$$

From the LFSR, two groups of parallel outputs are taken to be used as horizontal and vertical values. As can see in Fig. 7, the watermark data is replaced with the 4^{th} bit of the corresponding coefficient indicated by the two groups of parallel data from the LFSR. The 4th bit was chosen by considering the experimental results in the previous section.

3.2 Watermarking in MDWT-Domain

For the frequency domain, we considered only MDWT based on the previous experiments. Because of the property of a fringe pattern, the lowest subband after DWT is less sensitive to the spatial data change than others, as the experimental results. Also because the energy is concentrated to this subband, most coefficients have very high positive value. Thus, we chose the lowest subband as the watermarking places.

The watermarking scheme in DWT domain is shown in Fig. 8. Because we chose $1,024 \times 1,024$ $[Pixels^2]$ as the size of a fringe pattern and 32×32 $[Pixels^2]$ as the size of watermark, 5-level 2D MDWT is performed, which results in the size of the lowest subband the same as that of watermark. For DWT-domain watermarking, we selected a bit-plane replacement method. That is, the 3^{rd} lowest bit-plane is replaced with the whole watermark image data. Then the whole image is inversely 2D MDWTed to reconstruct the fringe pattern.

4 Experimental Results

To test the proposed watermarking scheme in the previous section, more than 100 fringe patterns were watermarked, attacked, and reconstructed. As mentioned

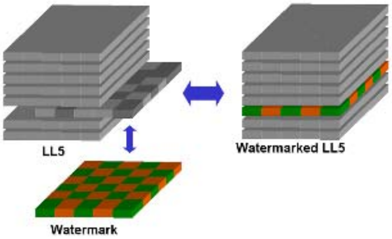


Fig. 8. Watermarking scheme in the MDWT domain

before, the size of a fringe pattern is $1,024 \times 1,024$ $[Pixels^2]$ and the size of watermark is 32×32 $[Pixels^2]$. To figure out the robustness and imperceptibility of the schemes, we considered the four kinds of attacks: JPEG compression, Gaussian noise addition, blurring, and sharpening attacks by Adobe Photoshop^{TM}.

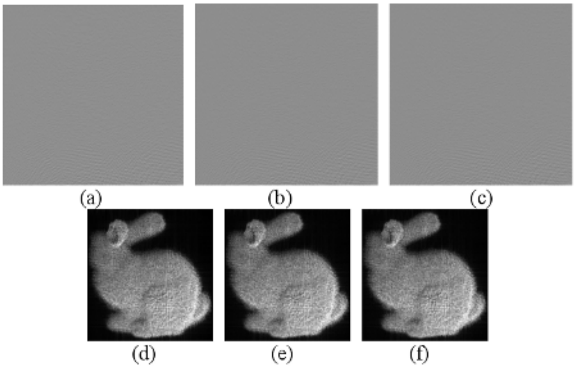


Fig. 9. Watermarking example for rabbit image; fringe pattern of (a) before, (b) after watermarking in spatial domain, (c) after watermarking in DWT domain; holographic image of (d) before, (e) after watermarking in spatial domain, (f) after watermarking in DCT domain

Fig. 9 shows examples of watermarked fringe patterns and the corresponding reconstructed holographic images for rabbit image. As can recognize from the figures, the three fringe patterns and the holographic images are indistinguishable from the original ones, which means the visual perceptibility of each scheme is very low. Table 3 shows the normalized correlation (NC) values of the watermarked fringe patterns and their reconstructed holographic images with respect to the original ones. In all cases the NC values are high enough as expected. In Table 4, the average error ratios of the extracted watermarks after the four kinds of attacks are listed. For JPEG compression, up to 0-quality compression is performed.

For Gaussian noise, up to 10% is added because the reconstructed image from the fringe pattern added more than 10% of Gaussian noise is degraded in the quality so that it is not worthy to be used in commercially.

In all the experimental cases, MDWT-domain scheme showed better performance than the spatial-domain scheme. The worst performance in MDWT-domain scheme was for the Gaussian noise attacks, but the error rate was only 3% even at the strongest attack of 10% addition. In the spatial-domain scheme, the JPEG compression showed the worst attacks. At JPEG quality 0, which corresponds to about 20:1 in the compression ratio, the error rate was more than 19%. That means the spatial-domain scheme is very weak in data compression and it is not useful if a data compression is accompanied.

Table 3. Average NC values of fringe pattern and holographic image

Domain	NC Values	
	Fringe pattern	*Holographic image*
Spatial	0.99997	0.99962
MDWT	0.99996	0.99989

Table 4. Experimental results by watermarking in spatial-domain and MDWT-domain

Attack(Adobe Photoshop^{TM})	Error rates(%)	
	Spatial	*DWT*
JPEG quality 6	0	0
JPEG quality 4	0.2	0
JPEG quality 2	14.1	0
JPEG quality 0	19.4	0
Gaussian noise addition(5%)	1.5	0
Gaussian noise addition(10%)	12.5	3.0
Sharpening	0	0
Blurring	0.4	0

For reference, Fig. 10 shows some examples of the extracted watermarks after attacks. In this figure, only the cases that the error ratio are 0.2%, 1.5%, 5.5%, 12.5%, and 19.4% are included but the other cases can be predicted from these result. As can imagine from the figures, an extracted watermark with error ratio of less than 10% is useful for the protection of ownership.

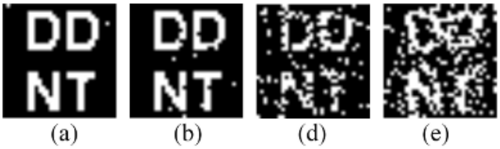


Fig. 10. Extracted watermarks: error ratio (a) 0.2, (b) 1.5, (c) 12.5, (d) 19.4

5 Conclusions

In this paper, we proposed a frequency-domain and a spatial-domain electronic watermarking schemes for digital holograms. The spatial-domain scheme was only for comparison. Here, we regarded the fringe patterns as the spatial-domain data and the results from mallat-tree 2D DWT are the frequency-domain data. The spatial-domain scheme disperses the watermark data throughout the fringe pattern. DWT-domain scheme embeds the watermark data by replacing a specific bit-plane of the lowest subband with the watermark data plane.

Experimental results indicated that both schemes are very imperceptible in fringe pattern and in the reconstructed holographic image. Also they showed quite high strength against most of the attacks. Generally they showed lowest robustness to the Gaussian noise addition attacks. But the spatial-domain scheme showed very weak on the data compression attacks. In all the cases, the spatial-domain scheme was worse than DWT-domain scheme and showed useless when data compression is accompanied. Thus, the DWT-domain watermarking scheme is expected to be used effectively to protect the ownership of digital hologram. Also, we expect that the contents in this paper can be the bases for further research on the electronic watermarking for digital holograms.

Acknowledgement

This work was supported by grant No. (R01-2006-000-10199-0) from the Basic Research Program of the Korea Science & Engineering Foundation.

References

1. Ingemar J. C., Matthew L. M., Jeffrey A. B.: Digital Watermarking, Morgan Kaufmann Publishers, San Francisco, CA (2002)
2. Ulf S. and Werner J.: Digital Holography, Springer Berlin Germany (2005)
3. Kishk S., Bahram J.: 3D Object Watermarking by a 3D Hidden Object. OPTICS EXPRESS, Vol. 11, No. 8 (2003) 874-888
4. Hyun K., Yeon L.: Optimal Watermarking of Digital Hologram of 3-D object. OPTICS EXPRESS, Vol. 13, No. 8 (2005) 2881-2886

Off-Line Signature Verification Based on Directional Gradient Spectrum and a Fuzzy Classifier

Young Woon Woo, Soowhan Han, and Kyung Shik Jang

Department of Multimedia Engineering, Dong-Eui University San 24, Gaya-Dong,
Pusanjin-Gu, Pusan, 614-714, Korea
ywwoo@deu.ac.kr

Abstract. In this paper, a method for off-line signature verification based on spectral analysis of directional gradient density function and a weighted fuzzy classifier is proposed. The well defined outline of an incoming signature image is extracted in a preprocessing stage which includes noise reduction, automatic thresholding, image restoration and erosion process. The directional gradient density function derived from extracted signature outline is highly related to the overall shape of signature image, and thus its frequency spectrum is used as a feature set. With this spectral feature set, having a property to be invariant in size, shift, and rotation, a weighted fuzzy classifier is evaluated for the verification of freehand and random forgeries. Experiments show that less than 5% averaged error rate can be achieved on a database of 500 samples including signature images written by Korean letters as well.

1 Introduction

Handwritten signature verification is concerned with determining whether a particular signature truly belongs to a person, so that forgeries can be deleted. Most of us are familiar with the process of verifying a signature for identification, especially in legal, banking, and other high security environments. It can be either on-line or off-line, which is differentiated by the data acquisition method[1]. In an on-line system, signature traces are acquired in real time with digitizing tablets, instrumented pens, or other specialized hardwares during the signing process. In an off-line system, signature images are acquired with scanners or cameras after the complete signatures have been written. There have been over a dozen prior research efforts and the summaries of these efforts are shown in [2].

However, most of the prior works on handwriting have used real-time input, which means they dealt with on-line system[3][4][5]. Relatively only a few methods focused on off-line signature verification. In off-line system, image of a signature written on a paper is obtained either through a camera or a scanner and obviously dynamic information is not available. Since the volume of information available is less, signature analysis using off-line techniques is relatively more difficult. To solve this off-line signature verification problems, elastic image matching techniques[6], extended shadow-coding method[7], and 2-D FFT(Fast

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1018–1029, 2006.

Fourier Transform) spectral method[8] were presented in the past. For the case when the actual (or true) signature and the forgery are very similar, Ammar et al. introduced an effective approach based on pressure features of the signature image[9]. More recent efforts tend to utilize a neural network classifier or a fuzzy classifier with the directional probability of the gradient on the signature image[10][11] and with geometric features for the detection of random and freehand forgeries[12]. And for the detection of skilled forgeries, Hidden Markov Model (HMM) is successively applied[13]. On reviewing the literature it was realized that a direct comparison of results from different researchers is often impossible. This is due to factors such as different data set used, field conditions, training and test data size, and the way in which the issue of forgery was handled[1][2]. Therefore, the main focus of this paper is to introduce a new technique for carrying out off-line signature verification and it is compared with our previous work[10] only. In this study, the global features based on FFT spectrum of directional gradient density function, which can abstract the overall shape information of signature image, and a weighted fuzzy classifier are proposed. The feature set extracted from directional gradient density function was developed earlier and widely used in off-line signature verification systems[10][11]. It is easy to extract and relatively well contains the overall shape information of signature image. However, the directional density function in [10][11] was derived from the entire signature image and easily affected by the rotation of image. Thus in this paper, the directional density function is extracted from only the outline of signature image not the entire signature image because the overall shape information is more densely located at the outline of image, and the FFT spectrum of that is utilized as a feature vector for invariance of image rotation and data reduction. The summary of our verification process is shown in Fig 1.

The design of a complete AHSVS (Automatic Handwritten Signature Verification System) which is able to cope with all classes of forgeries (random, freehand,

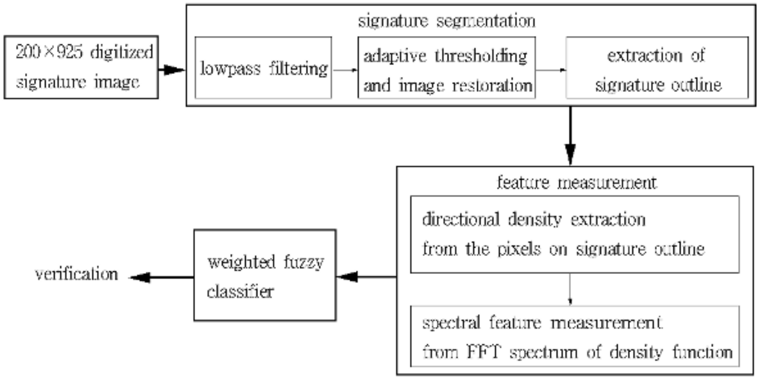

signature segmentation
200×925 digitized signature image
lowpass filtering
adaptive thresholding and image restoration
extraction of signature outline
feature measurement
directional density extraction from the pixels on signature outline
spectral feature measurement from FFT spectrum of density function
weighted fuzzy classifier
verification

Fig. 1. Overall processing steps for the proposed freehand or random forgery detection system

and traced [1]) is a very difficult task because of computational resources and algorithmic complexity. A better solution might be to subdivide the decision process in a way to eliminate rapidly gross forgeries like random or freehand forgeries. In this study, we focus on the construction of a first stage of verification system in a complete AHSVS. Thus, only the freehand forgeries, which are written in forgers' own handwriting style without knowledge of the appearance of genuine signature, or the random forgeries, which use his/her own signatures instead of genuine signatures, are considered.

2 Signature Image Segmentation and Feature Measurement

Signature image segmentation: The extraction of a signature image from the noisy background is done as follows. The first work is to apply a lowpass filter, shown in equation (1), to a scanned image for the noise reduction.

$$p'(i,j) = \frac{1}{9} \sum_{l=i-1}^{i+1} \sum_{k=j-1}^{j+1} p(l,k) \quad (1 \le i \le m, 1 \le j \le n) \tag{1}$$

where $p(l,k)$: the original image, $p'(i,j)$: the averaged image and m by n is the size of image (200 by 925). In a next, the threshold value, THD, is automatically selected from the averaged image, based on a simple iterative algorithm proposed by Ridler et al[14], and the fine signature image is restored as shown in equation (2).

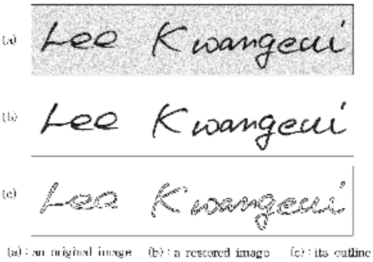
$$p''(i,j) = p(i,j) \quad if\, p'(i,j) > THD, \quad otherwise \quad p''(i,j) = 0 \tag{2}$$

where $p''(i,j)$: the restored image. In a third, the outline of signature is extracted from the restored image $p''(i,j)$ by using the erosion process. If $p''(i,j) > 0$ and its 8-neighbor count is below 8, $p''(i,j)$ must be on the outline of signature image, which is shown in equation (3).

$$\hat{p}(i,j) = p'(i,j) \quad if\, p''(i,j) > 0\, and\, Np < 8 \quad otherwise \quad \hat{p}(i,j) = 0 \tag{3}$$

where Np is a count of 8-neighbor pixels which are greater than zero. A sample signature image restored from noisy background and its outline is shown in fig. 2.

Feature Measurement: The one utilized as the input of a weighted fuzzy mean classifier for the verification process is the FFT spectral feature vector of directional gradient density function extracted from only the outline of signature image. It depends on the overall shape of the signature image, and so is assumed to have enough information for the detection of freehand or random forgeries. In the gradient computation process, Sobel 3 by 3 mask shown in fig. 3 is convolved with each pixel on the restored image if and only if it is on the outline of signature image, and the amplitude and angular orientation of gradient vector are computed by equation (4) and (5).



(a) : an original image (b) : a restored image (c) : its outline

Fig. 2. A sample signature restored from noisy background and its outline

$$\begin{array}{cc} \textit{row gradient} & \textit{column gradient} \\ S_r = \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} & S_c = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \end{array}$$

Fig. 3. Sobel 3 by 3 gradient mask

$$\text{if and only if } \hat{p}(i,j) > 0, \quad Gr(i,j) = Sr(i,j) \otimes p''(i,j)$$

$$Gc(i,j) = Sc(i,j) \otimes p''(i,j) \quad and, \quad G(i,j) = \sqrt{Gr(i,j)^2 + Gc(i,j)^2} \tag{4}$$

$$\theta(i,j) = \left[\tan^{-1}\left(\frac{Gc}{Gr}\right) + \frac{\pi}{2}\right] * \frac{128}{\pi} \tag{5}$$

where $\hat{p}(i,j)$: a pixel on the outline of signature image and $p''(i,j)$: a pixel on the restored image. The multiplication term, $\frac{128}{\pi}$ in equation (5), allows the angular orientation $\theta(i,j)$ to have a range from 0 to 127 for FFT. In a next, the directional gradient density function for the pixels on the outline of signature image, $DF(\theta_k)$, is derived by equation (6) and its normalized term, $NF(\theta_k)$, by equation (7).

$$DF(\theta_k) = \sum_{i=1}^{200}\sum_{j=1}^{925} X(\theta_k, i, j) \tag{6}$$

where $X(\theta_k, i, z) = G(i,j), \theta(i,j) = \theta_k$, derived by equations (4),(5), and k=0, 1 ,... 127.

$$NF(\theta_k) = \frac{DF(\theta_k)}{\sum\limits_{\theta_k=0}^{127} DF(\theta_k)} \tag{7}$$

Finally, as a means of data reduction and feature selection, the 128 point FFT is taken into the normalized directional gradient density, which is shown in equation (8).

$$F(m) = abs\left(\sum_{\theta_k=0}^{127} NF(\theta_k)\exp(-2\pi j\theta_k m/128)\right) \tag{8}$$

where m=0,1,2,....,127. A D.C. component, $F(0)$, is always zero because the mean value of $NF(\theta_k)$ is removed before FFT. Thus, in this study, the first 15 spectral components except $F(0)$ are utilized as a feature vector to be an input of the weighted fuzzy classifier for verification. The feature vector formed with $F(1), F(2), ..., F(15)$ has a property to be invariant in size, shift, and rotation. Fig. 4 shows some samples of genuine and forged signatures written by Korean letters and their feature vectors. The FFT spectral feature vectors extracted from two genuine signatures, (a) and (b) in Fig. 4., even one of them is scaled and rotated, are very similar together, but different with feature vectors extracted from two kinds of freehand forgeries written by two different person, (c) and (d) in Fig 4.

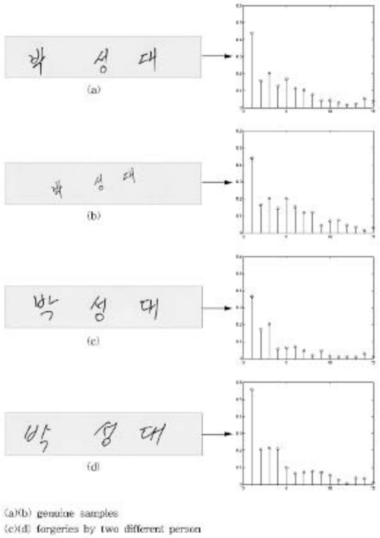


Fig. 4. Samples of genuine and forged signatures written by Korean letters and their FFT spectral feature vectors

3 A Weighted Fuzzy Classifier

The construction of a fuzzy classifier depends on the type of a fuzzy membership function and the calculation method of a fuzzy mean value[15]. The triangular type of membership function has a simple configuration and is easy to apply where the only one reference feature set for one target pattern is used as in our experiments. Thus the one used in this paper as a classifier is combined a triangular fuzzy membership function with a weighted fuzzy mean method which utilizes the variances of each dimensional feature value in a reference feature set as the weights, ω_i. This is shown in equation (9).

$$h_w(\mu_1(x_1), \mu 2(x_2), \cdots, \mu_n(x_n); \omega_1, \omega_2, \cdots, \omega_n) = \sum_{i=1}^{n} \mu_i(x_i) \cdot \omega_i, (\sum_{i=1}^{n} \omega_i = 1) \quad (9)$$

where h_w: a weighted fuzzy mean value, μ_i: a membership grade extracted from a triangular membership function, w_i: a weight for an i^{th} feature value, x_i , and n is the dimension of incoming feature vector (15 in this paper).

This type of classifier does not require a training stage while the classifier based on neural network algorithms does. And the performance of a neural classifier highly depend on its architecture and learning algorithm. In this fuzzy classifier, the triangular fuzzy membership functions of each of fifteen-dimensional feature values and their variances used as weights are simply constructed and computed by using the reference feature set, and utilized for the verification of a signature image without any further process. Thus the evaluation process is much simpler and easier than that of the conventional neural network classifier. For an incoming test signature, the fifteen membership grades are computed by pre-established triangular membership function, and its weighted fuzzy mean value is derived by equation (9). If it is greater than a threshold value, this signature is verified as a genuine signature, and if not, it is discarded as a forgery. More details about this fuzzy classifier based on triangular membership function and weighted fuzzy mean value can be founded in our previous work[10].

4 Experiments and Performance Assessment

In our experimental process, a total of 500 samples including signature images written by Korea letters were corrected and verified by both of the proposed method and our previous algorithm[10] where the twelve-dimensional directional density extracted from an entire signature image had been utilized as a feature vector. Test images belong to five sets of different signatures. Signature data collection for each set was done by as follows. One of five different writers was chosen as a target and asked to write his own name twenty times on an A4 page, and four of the remaining writers were assigned to be forgers. Each of the forgers was asked to write the targeted name twenty times in his own handwriting on an A4 page. The forgers were not allowed to study the samples of the original signature because this study focused on only freehand or random forgery detection not skilled forgery detection. Each set comprises one A4 page of genuine and four A4 pages of freehand forgery signatures. They were scanned, one page at a time, at resolution of 300 dpi, 8-bit gray-scale, and each signature was stored in a 200 by 925 pixel matrix. Thus each set contains 20 genuine signatures and 80 freehand forgeries. By use of each of five writers' own signature as a target, five sets of different signature classes were made. Some samples of data set 1 to 5 are shown in fig. 5, and those of data set 2 and 5 are written by Korean Letters.

Signature verification with a neural classifier needs the variety of forged signatures to train the classifier for the high performance[16]. However, under the real world environment, only a few forged signature samples are available. In this study, a fuzzy mean classifier without any knowledge of forged signatures is presented to decide an incoming signature whether it belongs to a genuine or forged signature. The construction of a fuzzy classifier and the verification process were done by as follows. In a first, the reference feature set is constructed

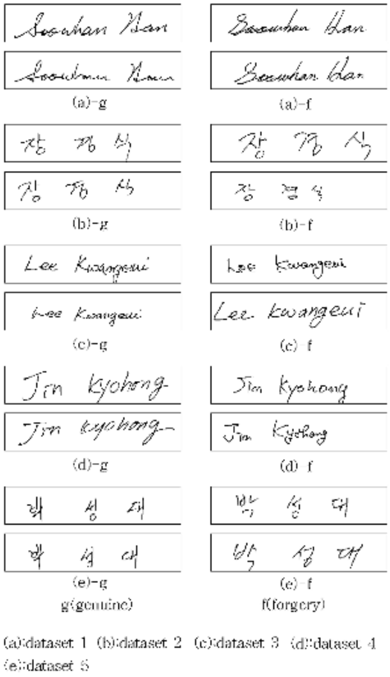


Fig. 5. Sample signatures in data set 1 to 5

only with the randomly selected genuine signature samples shown in equation (10), and the weights for each of fifteen-dimensional feature values are derived by equation (11).

$$rf(x_i) = \frac{1}{nm}\sum_{j=1}^{nm} f_j(x_i); i = 1, 2, ..., 15 \tag{10}$$

where nm : the number of selected genuine signature samples, $f_j(x_i)$ is an i^{th} feature value of feature set, f_j, extracted from a signature sample j, and $rf(x_i)$ is an i^{th} feature value of reference feature set. For the weights, the normalized variances for each of fifteen-dimensional feature values, $vr_1, vr_2, ..., vr_{15}$, are drived as in our previous work[10]. And if a normalized variance for the i^{th} feature values extracted from the selected signature samples, vr_i, is a p^{th} larger value among $[vr_1, vr_2, ..., vr_{15}]$, then a weight for the i^{th} feature value is defined as

$$\omega_i = a(16-p)^{th} \quad l\arg er \quad value \quad in \quad [vr_1, vr_2, ..., vr_{15}] \tag{11}$$

where a is a small constant.

By the equation (11), the i^{th} feature which has a smaller variance has a larger weight, and it means the i^{th} feature whose values are not significantly changed between genuine signature samples is more weighted in verification process. After this stage, the verifier performance is evaluated. For an incoming signature image, the membership grades of its feature values are derived by equations (12)-(14).

$$\mu_i(x_i) = \frac{(x_i - rf(x_i))}{rf(x_i)} + 1 \quad if \quad x_i < rf(x_i) \tag{12}$$

$$\mu_i(x_i) = -\frac{(x_i - rf(x_i))}{rf(x_i)} + 1 \quad if \quad x_i \geq rf(x_i) \tag{13}$$

$$\mu_i(x_i) = \begin{matrix} \mu_i(x_i) \; if \; \mu_i(x_i) \geq 0 \\ 0 \quad if \; \mu_i(x_i) < 0 \end{matrix} \tag{14}$$

where x_i is an i^{th} feature value of input signature image, $rf(x_i)$ is an i^{th} feature value of reference feature set, and $\mu_i(x_i)$ is a membership grade for x_i. All of membership functions are configured as a triangular type shown in figure 6.

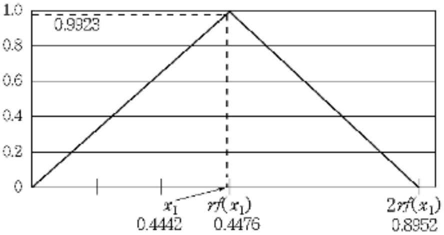


Fig. 6. A fuzzy membership grade, $\mu_1(x_1)$, for a signature sample shown in fig.4-(a)

In a next, the weighted fuzzy mean value is derived by equation (15) and the signature is verified by equation (16).

$$h(\mu_1(x_1), \mu 2(x_2)), \cdot\cdot\cdot, \mu_{15}(x_{15}); \omega_1, \omega_2, \cdot\cdot\cdot, \omega_{15}) = \sum_{i=1}^{15} \mu_i(x_i) \cdot \omega_i \tag{15}$$

where h is the weighted fuzzy mean value of an incoming signature image, μ_i is a membership grade of the i^{th} feature value, and w_i is a weight shown in equation (11).

$$\begin{matrix} h & \geq & Th & \text{accepted as a genuine signature} \\ h & < & Th & \text{rejected as a forged signature} \end{matrix} \tag{16}$$

where Th is a threshold value. The particular value of Th will determine the probabilities of false rejection(FRR: Type I error) and false acceptance(FRA: Type II error). The choice of Th should therefore be based on the cost of these two types of errors. In our experiments, Th is expressed by equation (17), and selected to minimized the verification error, ERR, defined by equation (18).

$$Th = mh - nh(1 - \frac{K}{sh}) \tag{17}$$

where mh, nh and sh are mean, minimum and standard deviation of the weighted fuzzy mean values in equation (15), respectively, which are derived by the selected genuine signature samples, that were used for the construction of reference feature set, and K is a constant.

$$ERR(\%) = \frac{FRR + FRA}{2} \tag{18}$$

ERR is computed with the changes of constant K, and Th that corresponds to K's value which gives the minimum ERR is chosen as the pre-established Th. Fig. 7 shows the relation between FRR, FRA, and ERR with the different values of K, which determines Th. ERR is usually smallest around the crossing point of FRR and FRA curves, which means Th is selected where type I and type II error rates do not have a significant difference. And it is also shown in fig. 7 that selecting Th is not critical because ERR is less than 10% for a wide range of K.

In the experiments, the performance of weighted fuzzy classifier was evaluated for both random and freehand forgeries. Additionally, it was checked with the twelve-dimensional directional feature set from [10] as well for the comparison purpose. For each case, five independent simulations with a different choice of five randomly selected genuine signature samples for the construction of reference feature set were performed, and the verification results were averaged. Under random forgery test, 420 signature samples(20 genuine signatures and 400 random forgeries) for each of five signature classes were evaluated. The average ERR for all of signature classes is just below 4.1% by the proposed method and 5.7% by the algorithm in [10]. They are summarized in table 1 for each signature class. Under freehand forgery test, the test size for each of five signature classes is 20 genuine samples and 80 freehand forgery samples written by four different forgers, and the average ERR with the proposed feature set and with the directional feature from [10] were about 5.3% and 9.6% respectively. They are summarized in table 2 for each signature class.

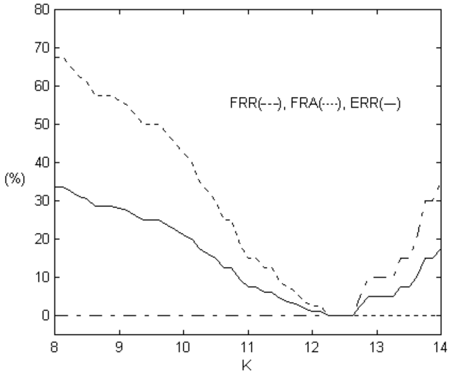


Fig. 7. The relation between FRR, FRA and ERR with different values of K

From table 1 and 2, it is obvious that the performances by both algorithms with random forgery are better than with freehand forgery because the feature vectors used in both of two approaches are based on the overall shape of signature images. In random forgery test, the difference of ERR by two algorithms is not significant even though the ERR by the proposed method is always lower for each signature class. However, the proposed method shows much better results in freehand forgery test, which means the directional feature set extracted from the outline of signature image has more accurate shape informations than the feature set from the entire signature image does. The verification process between similar signature images, such as freehand forgery test, always requires more precise shape information.

The overall experimental results show that the weighted fuzzy classifier with the feature vector extracted from FFT spectrum of directional gradient density function of signature outline is relatively effective in both of random and freehand forgery detection, even the signature image is written by Korean letters. This proposed method has rotation invariant characteristic and preserves more accurate shape information, and thus it can be possibly applied as a first stage verifier in off-line signature verification system.

Table 1. Average verification results for random forgery test after five independent simulations.($FRR = \frac{\# of\, false\, rejected\, signature}{20} \times 100$, $FRA = \frac{\# of\, false\, accepted\, signature}{400} \times 100$, $ERR = \frac{FRR+FRA}{2}$).

	FRR		FRA		ERR	
	proposed method	algorithm in [10]	proposed method	algorithm in [10]	proposed method	algorithm in [10]
data set 1	2%	4%	1.8%	3%	1.9%	3.5%
data set 2	5%	5%	10%	12%	7.5%	8.5%
data set 3	1%	2%	10.25%	13%	5.63%	7.5%
data set 4	2%	4.7%	7.9%	9.5%	4.95%	7.1%
data set 5	0%	1%	0.5%	2%	0.25%	1.5%
Average	2%	3.34%	6.09%	7.9%	4.05%	5.62%

Table 2. Average verification results for freehand forgery test after five independent simulations.($FRR = \frac{\# of\, false\, rejected\, signature}{20} \times 100$, $FRA = \frac{\# of\, false\, accepted\, signature}{80} \times 100$, $ERR = \frac{FRR+FRA}{2}$).

	FRR		FRA		ERR	
	proposed method	algorithm in [10]	proposed method	algorithm in [10]	proposed method	algorithm in [10]
data set 1	2%	5.6%	0.75%	4%	1.38%	4.8%
data set 2	7%	10%	13.75%	19%	10.38%	14.5%
data set 3	5%	9.2%	7.5%	11.5%	6.25%	10.35%
data set 4	4%	10%	9%	17.6%	6.5%	13.8%
data set 5	3%	5%	1.25%	4%	2.13%	4.5%
Average	4.2%	7.96%	6.45%	11.22%	5.33%	9.59%

5 Conclusions

An off-line signature verification method based on spectral feature extraction of signature outline and weighted fuzzy classifier is described, and its effectiveness is evaluated with 500 signature samples. In our experiments, only the genuine signature samples are utilized for the construction of reference feature because a few forged signature samples are available under the real world environment, and the training period is not required in this type of classifier. From the high performance results, it is known that the proposed system detects relatively well the random or freehand forgeries, and thus it can be utilized as a first stage verifier which can help to improve both the speed and accuracy of the complete off-line signature verification system. The further research should involve the evaluation with a larger data set written by more varied writers for the real world applications and the investigation of skilled forgeries detection system for the complete off-line signature verification system.

References

1. R. Plamondon and G. Lorette, "Automatic signature verification and writer identification-the state of the art," *Pattern Recognition*, Vol.22, No.2, pp. 107-131, 1989.
2. R. Palmondon and S. Srihari, "On-line and Off-line Handwriting Recognition: A Comprehensive Survey", *IEE Trans. on Pattern Analysis and Machine Interlligence*, Vol.22, No.1, pp.63-83, 2000.
3. D.U. Cho and Y.L. Bae, "The modified DTW method for on-line automatic signature verification", *Journal of Korea Information Processing Society*, Vol.10-B, No.4, pp.451-458, 2003.
4. W.S. Lee and S.H. Kim, "An on-line signature verification algorithm based on neural network", *Journal of Korea Intelligent Information Systems*, Vol.7, No.2, pp. 143-151, 2001.
5. M. Fuentes, S. Salicetti and B. Dorizzi, "On-line signature verification: Fusion of a hidden Markov model and a neural network via a support vector machine", 8^{th} *International Workshop on Frontiers in Handwriting Recognition(IWFHR'02)*, pp.253, Aug. 2002, Canada
6. F. Nouboud, *Handwritten signature verification: a global approach, Fundamentals in Handwriting Recognition,* S, Impedovo, ed., pp.455-459, Springer, Berlin, 1994.
7. R. Sabourin, M. Cheriet and G. Genest, " An extended shadow-code based approach for off-line signature verification", *Proc. ICDAR'93: Int. Conf. on Document Analysis and Recognition,* Tsukuba Science City, Japan, pp.1-5, IEEE Computer Society Press, 1993.
8. S.N. Nam, J.Y. Park, and S.B. Rhee, "A Study on Signature Identification using the Distribution of Space Spectrum", *Journal of the Korea Institute of Telematics and Electronics,* Vol.30, No.9, pp.638-644, August, 1993, Korea.
9. M. Ammar, Y. Yoshida, and T. Fukumura, "A new effective approach for off-line verification of signatures by using pressure features," *Proc. 8th Int. Conf. on Pattern Recognition,* pp.566-569, 1986.
10. S.W. Han and J.K. Lee, "A Study on Off-Line Signature Verification using Directional Density Function and Weighted Fuzzy Classifier", *Journal of Korea Multimedia Society*, Vol.3, No.6, pp.592-603, Dec., 2000.

11. M. Hanmandlu, K.R. Mohan, S. Chakraborty and G. Garg, "Fuzzy modeling based signature verification system", 6^{th} *International conference on document analysis and recognition (ICDAR '01)*, pp.110, Sept., 2001, WA. U.S.A.
12. H. Kai and H. Yan, "Off-Line Signature Verification Based on Geometric Feature-Extraction and Neural Network Classification", *Pattern Recognition Letters,* Vol.30, No.1, pp.9-17, 1997.
13. E. Justino, F. Bortolozzi and R. Sabourin, "Off-line signature verification using HMM for random, simple and skilled forgeries", 6^{th}*international conference on Document analysis and Recognition(ICDAR '01),* pp. 1031, 2001, Seattle, WA, U.S.A.
14. Ridler and Calvard, "Picture thresholding using an iterative selection method", *IEEE Transactions on Systems, Man and Cybernetics*, Vol.8, No.8, pp.630-632, 1978.
15. H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, pp. 217-239, Kluwer Academic Publishers, 1996.
16. F.M. Ham, I. Kostanic, *Principles of Neurocomputing for Science and Engineering*, New York: McGraw-Hill, 2001.

Automatic Annotation and Retrieval for Videos

Fangshi Wang[1,2], De Xu[1], Wei Lu[2], and Hongli Xu[1]

[1] School of Computer & Information Technology, Beijing Jiaotong University, Beijing China
[2] School of Software, Beijing Jiaotong University, Beijing China 100044
{fshwang, dxu, wlu, hlxu}@bjtu.edu.cn

Abstract. Retrieving videos by key words requires semantic knowledge of the videos. However, manual video annotation is very costly and time consuming. Most works reported in literatures focus on annotating a video shot with either only one semantic concept or a fixed number of words. In this paper, we propose a new approach to automatically annotate a video shot with a varied number of semantic concepts and to retrieve videos based on text queries. First, a simple but efficient method is presented to automatically extract Semantic Candidate Set (SCS) for a video shot based on visual features. Second, a semantic network with *n* nodes is built by an Improved Dependency Analysis Based Method (IDABM) which reduce the time complexity of orienting the edges from $O(n^4)$ to $O(n^2)$. Third, the final annotation set (FAS) is obtained from SCS by Bayesian Inference. Finally, a new way is proposed to rank the retrieved key frames according to the probabilities obtained during Bayesian Inference. Experiments show that our method is useful in automatically annotating video shots and retrieving videos by key words.

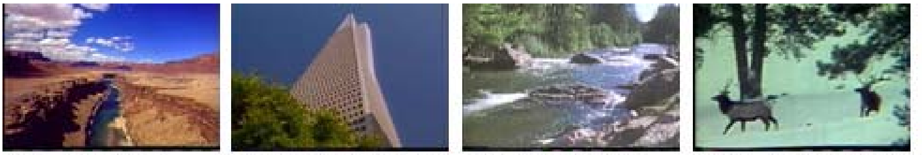
1 Introduction

The increasing amount of multimedia information is driving the demand for content-based access to video data. Queries based on the low-level video feature like color or texture have been proposed for retrieving videos by content, but most users find it difficult to query using such visual attributes. Most people would prefer to pose text queries and find videos relevant to those queries. For example, one should be able to pose a query like “cars on a road”. This needs to bridge the semantic gap between the low-level feature descriptions and the semantic descriptions of multimedia. Manually annotating videos is not only labor intensive and time-consuming for large video archives, but also subject to human errors [1]. Automatic video annotation is becoming more and more useful for improving the performance of multimedia information retrieval.

In most previous research works, the task of annotating a non-annotated video can be viewed formally as a classification problem, we must make a yes/no decision for each word in the vocabulary [2][3][4]. Experts or users specify several classes, system can construct one or several classifiers through learning from the training set which is built manually by users. The visual features of a new video are extracted automatically and input into the well-trained classifiers. Then the result of the classification is the semantic annotation of the new shot. The classes defined in such method are mutually exclusive, so each video shot can have only one semantic concept.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1030 – 1040, 2006.

In fact, one concept is not enough to fully summarize a video shot with rich contents. A shot could include more than one concept. For example, three concepts, such as “land”, “sky” and “cloud”, are needed to describe the first picture of Figure 1. The frame should not alone belong to any one of the three classes. Obviously, semantic concepts do not occur independently or are not isolate from each other, and the mutual information between them should be taken into account in order to make the annotation complete.



land, sky, cloud building, greenery, sky water, greenery animal, snow, greenery

Fig. 1. A random selection of key frames from the training set

Intuitively it is clear that the presence of a certain concept suggests a high possibility of detecting certain other concepts. Similarly some concepts are less likely to occur in the presence of others. The detection of “car” boosts the chances of detecting “road”, and reduces the chances of detecting “waterfall”. It might also be possible to detect some concepts and infer more complex concepts based on their relation with the detected ones. Naphade [5] proposed the MultiNet as a way to represent higher level probabilistic dependencies between concepts. However, both the classes and structure of the classification frameworks were either decided by experts or specified by users. Moreover, the structure will become very large with the increase of the class number. If there are n classes, there will be n variable nodes and $n(n\text{-}1)/2$ function nodes and $n(n\text{-}1)$ edges in the MultiNet.

More general approaches attempt to annotate new key frames with concepts in the annotations of training set. MediaNet[6] can automatically select the salient classes from annotated images and discover the relationship between concepts by using external knowledge resources from WordNet. However, the relationships between concepts in MediaNet are too complex. There are not only perceptual relationships such as "equivalent", "specializes", "co-occurs", and "overlaps", but also semantic relationships such as “Synonymy / Antonymy“, “Hypernymy / Hyponymy”, “Meronymy / Holonymy”, “Troponymy”, “Entailment“, which are summarized into a small subclass of all these relationships by clustering subsequently. In his summarized MultiNet, there is only "specializes" relationship such as “man” is a subtype of “hominid”. His main attention is on analyzing the sense of a word and generating the "specializes" relationship and so on.

In the application of annotating the video, we only concern about whether concept B is also present in the same frame if concept A is present. So we want to discovery the coexisting relationship among multiple concepts.

Simpler methods are proposed to automatically annotate videos with a fixed number of concepts [7]. But it is not reasonable to label every shot with a fixed number of concepts, no matter whether the shot content is rich or not.

In order to overcome the above shortcomings, a new method is proposed to annotate a video shot with a varied number of concepts. This paper is organized as follows. A simple but efficient approach is proposed to extract Semantic Candidate Set (SCS) in section 2. An improved DABM algorithm is proposed to construct the semantic network in section 3. Section 4 describes a way to select the final annotation set (FAS) from the SCS by Bayesian Inference and automatically annotate a video shot with a varied number of concepts. Section 5 introduces a probabilistic method to rank the retrieved frames based on Bayesian Inference. The experimental results are given in section 6. Finally, section 7 concludes the paper.

2 Extracting the Semantic Candidate Set for a New Shot

The Semantic Candidate Set (SCS) is a set of N most probable concepts according to the visual features. It is supposed that all concepts actually annotated for a testing frame are in its SCS, then the actual concepts are chosen from the SCS by Bayesian Inference.

A training set is constructed by manually annotating the key frames of videos shots and regarded as Ground Truth (GT). There is an annotation with 1-4 concepts for each key frame in the training set. Examples of selected key frames with annotation are shown in Figure 1**.** Each concept is considered a semantic class. Without loss of generality, suppose there are n semantic concepts in the training set. A simple method is introduced to obtain the SCS.

First, the center of each semantic class is calculated as follows.

$$C_{S,k} = \frac{1}{|T_S|}\sum_{f_j \in T_S} f_{j,k} \qquad (k\ \ 0,\ldots,\dim-1) \tag{1}$$

where *dim* is the dimension of the visual feature vector, T_S is the set of the samples with concept S in their semantic annotation, $/T_S|$ is the number of the samples in T_S, f_j is the *j*th frame in T_S , $f_{j,k}$ is the *k*th visual feature element of frame f_j and $C_{S,k}$ the *k*th visual feature element of the class center of concept S.

Then, formula (2) is used to compute the distances between the key frame F of a new shot and every semantic class center, which are denoted as *Dist*[1], *Dist*[2],…, *Dist*[*n*].

$$Dist[i] = \sqrt{\sum_{k=0}^{\dim-1} (F_k - C_{i,k})^2} \quad (i = 1, \ldots, n) \tag{2}$$

where F_k is the *k*th visual feature element of the new key frame.

Finally, *Dist*[1,…,*n*] is sorted from small to large. N most probable concepts, denoted as S1,…, SN corresponding the N smallest distances, consist of SCS. In our experiment, the result is the best for $N = 4$. S1 is regarded as the first concept of the new shot. This process of extracting SCS is named Semantic Class Centre Method (SCCM).

3 Constructing a Semantic Network

Bayesian Network (BN) is a graphical model that efficiently encodes the joint probability distribution over a set of random variables. BN is selected to construct the Semantic Network, in which a node represents a semantic concept and an edge represents the dependency relationship between two concepts. Benítez [6] gave two reasons to select Bayesian networks to learn statistical dependencies between concepts.

Learning of Bayesian network includes two parts: learning the structure and learning the parameters given a structure. A three-phase construction mechanism is representative of Dependency Analysis Based Method (DABM) and is used to construct Bayesian Network [8]. Since a structure encodes many dependencies of the underlying model, the algorithms try to discover the dependencies from the data, then use these dependencies to infer the structure. The dependency relationships are measured by using kinds of CI test.

Mutual information and conditional mutual information are used as measurement. In information theory, mutual information is used to represent the expected information gained on sending one symbol and receiving another. The mutual information between two nodes can tell us whether two nodes are dependent and how close their relationship is. The mutual information of two nodes X,Y is defined as

$$\mathrm{I(X,Y)}=\sum_{x=0}^{1}\sum_{y=0}^{1}P(x,y)\log\frac{P(x,y)}{P(x)P(y)} \tag{3}$$

and the conditional mutual information is defined as

$$\mathrm{I(X,Y|C)}=\sum_{x=0}^{1}\sum_{y=0}^{1}\sum_{c=0}^{2^{|C|}-1}P(x,y,\mathrm{c})\log\frac{P(x,y\mid c)}{P(x\mid c)P(y\mid c)} \tag{4}$$

where 1 means presence, 0 means absence and *C* is a cut set of nodes. If I(X,Y) is smaller than a certain threshold , then X,Y are marginally independent. If I(X,Y|C) is smaller than the threshold, then X,Y are conditionally independent given *C*.

Cheng *et al.*[8] proposed the three-phase (drafting, thickening and thinning) construction mechanism which can add an edge between each pair of nodes depending or conditionally depending. They developed two efficient algorithms, Algorithm A and Algorithm B, based on the three-phase construction mechanism. Algorithm A deals with a special case where the node ordering is given to orient the edges, this algorithm only require $O(n^2)$ CI tests. However, it is not always easy for users to give a node ordering. Some of the sequences among concepts are easy to be determined, such as “sky” and “cloud”. The presence of “cloud” in a frame makes sure of the presence of “sky”, but otherwise it is not true. It means that “cloud” should be ordered before “sky” and the direction of the edge should be from “cloud” to “sky”. However, as for other concepts, such as “water” and “animal”, it is difficult even for an expert to determine their sequence. Algorithm B deals with the general case where the node ordering is not given and requires $O(n^4)$ conditional independence (CI) tests to orient the edges.

It concluded that Algorithm B was not as good as Algorithm A because Algorithm B did not use the node ordering as prior knowledge as Algorithm A did [8].

A new method is proposed to orient the edges in the general case where the node ordering is not given, and only requires O(n^2) instead of O(n^4) CI tests required by the traditional DABM (TDABM) for determining the directions of the edges.

Suppose the node $n1$ and $n2$ are dependent or conditionally dependent, an edge between them should be added. A scoring function is designed to orient the edge ($n1$, $n2$) as follows.

$$f(n_1,n_2)=\frac{score(n_1,n_2)}{score(n_2,n_1)}=\frac{P(n_2\mid n_1)\times P(\sim n_2\mid n_1)\times P(n_2\mid\sim n_1)\times P(\sim n_2\mid\sim n_1)}{P(n_1\mid n_2)\times P(\sim n_1\mid n_2)\times P(n_1\mid\sim n_2)\times P(\sim n_1\mid\sim n_2)} \tag{5}$$

where $score(n_1, n_2)$ represents the probability of node n_1 with node n_2 as its child and $score(n_2, n_1)$ the probability of node n_2 with node n_1 as its child. Applying the Bayesian formula to formula (5), then

$$f(n_1,n_2)=\frac{score(n_1,n_2)}{score(n_2,n_1)}=\frac{P(n_2)^2P(\sim n_2)^2}{P(n_1)^2P(\sim n_1)^2} \tag{6}$$

If the value of equation (6) is larger than 1.0, it indicates that the probability of n_1 with n_2 as its child is larger than that of n_2 with n_1 as its child, then the direction is from n_1 to n_2, otherwise from n_2 to n_1. The time complexity of orienting one edge is O(1) because the probability of each node has been calculated while calculating the mutual information using equation (3), and there are $n(n$-1)/2 edges at most in BN. So in the worst case, the time complexity of orienting all edges is O(n^2) in our method instead of O(n^4) in [8].

Having constructed the semantic network, the parameters, i.e. the conditional probability of each node, are learned by standard statistic method given the BN structure. We will give the detailed comparison between IDABM and the traditional algorithms (TDABM) in section 6.

4 Obtaining the Final Annotation Set by Bayesian Inference

Having obtained the semantic candidate set (SCS) and the first concept S1 of the new shot, we should have a way to determine which of the others in SCS are also present in the same shot. Bayesian inference is used to calculate the conditional probabilities of the other concepts given S1. Suppose that the initial current evidence set is CE={S1=1} (1 means presence, 0 means absence). If $P(S2=1|CE)>\sigma$, then S2 will be assigned to the new shot; If $P(S3=1|CE)>\sigma$, then S3 will be assigned to the new shot, and so on. We obtain the final annotation set of the shot by dynastic Bayesian inference. The procedure is similar to that proposed by Huang [9], as follows.

Step 1. The directed graph of Bayesian Network is moralized and triangulated into a chordal graph.

Step 2. A join tree (JT) is built from the chordal graph, which consists of cliques node and separator sets (abbreviated as sepset).

Step 3. Initialize the belief potential of each clique and sepset according to the conditional probabilities of the nodes in Bayesian Network.

Step 4. Select the final annotation set from the SCS as follows.

```
    NE={S1};         CE =∅;         SCS ={S1,…,SN}
    While (NE is not empty) do
     {Input NE into JT and modify the potentials in the
  JT;
      Perform Global propagation to make the potentials
      of JT locally consistent;
      $CE = CE \cup NE$ ;      $NE = \varnothing$;
      for (each concept Si in SCS) do
      if ((Si$\notin$ CE) and (P(Si=1|CE)>$\sigma$)) then
        { NE=NE $\cup$ {Si};   SP[f][Si]=P(Si=1|CE); }
     }
```

where *CE* is the final annotation set (FAS) of a shot after the procedure stops, and SP[*f*][Si] is used to store the probability of annotating the frame *f* with concept Si.

5 Ranked Retrieval Based on Annotation

The task of video retrieval is similar to the general ad-hoc retrieval problem. We are given a text query $Q=\{w_1, w_2, \ldots, w_k\}$ and a collection *V* of key frames of videos. The goal is to retrieve the video key frames that contain concept set *Q* in *V*.

A simple approach to retrieving videos is to annotate each key frame in *V* with a small number of concepts using the techniques proposed in section 2, 3 and 4. We could then index the annotations and perform text retrieval in the usual manner. This approach is very straightforward. However, it has a disadvantage that does not allow us to perform ranked retrieval. This is due to the binary nature of concept occurrence in automatic annotations: a concept either is or is not assigned to the key frame. When the annotations of many retrieved frames contain the same number of concepts, document-length normalization will not differentiate between these key frames. As a result, all frames containing the same number of concepts are likely to receive the same score.

Probabilistic annotation can be used to rank the relevant key frames (here 'relevant key frames' are the ones that contain all query concepts in their ground-truth annotation). A technique has been developed to assign a probability $P(S=1|CE)$ to every concept *S* in the annotation in section 4. The key frame is scored by the probability that a query would be observed. Given the query $Q= \{w_1, w_2, \ldots, w_k\}$ and a frames $f \in V$, the probability of containing *Q* in frame *f* is:

$$P(Q|f)=\prod_{j=1}^{k} P(w_j|f) \tag{7}$$

where $P(w_j \mid f)$ has already been computed and stored in $SP[f][w_j]$ in section 4. all retrieved frames are ranked according to $P(w_j \mid f)$ from large to small.

6 Experimental Results

We have chosen videos of different genres including landscape, city and animal from website www.open-video.org to create a database of a few hours of videos. Data from 35 video clips has been used for the experiments.

Table 1. The presence frequency of each concept in training set (%)

concept	**car**	**road**	**bridge**	**building**	**waterfall**	**water**	**boat**
percentage	3.08	6.84	2.66	19.39	1.14	42.97	6.08
concept	**cloud**	**sky**	**snow**	**mountain**	**greenery**	**land**	**animal**
percentage	13.31	44.87	6.84	14.45	32.7	25.48	25.48

First, we partition every video clip into several shots and to manually annotate the shots. The key frames of each shot are extracted automatically to form the samples set. The perceptual features such as HSV accumulated Histogram and Edge Histogram are extracted automatically from the sample set and stored into a video database after being normalized. In our experiments, there are 544 key frames for training and 263 key frames for testing. Fourteen different concepts are extracted automatically from the training set, which are shown in table 1.

6.1 Results: Automatic Videos Annotation

In this section we evaluate the performance of our method on the task of automatic video annotation. We are given an unannotated key frame *f* and are asked to automatically produce an annotation. The automatic annotation is then compared to the manual ground-truth annotation (GT).

It is said in [6] that different classifiers were evaluated including k-nearest neighbors, one-layer neural network, and mixture of experts, of which k-nearest neighbor was shown to outperform the rest. So we compare our method of selecting the semantic candidate set described in section 2 with K Nearest Neighbor (KNN) and Naïve Bayesian (NB) classifier. Four most probable concepts are also chosen for KNN and NB to build their SCSs. Three standards are used to measure the performance of the three methods as follows.

$$S1\_first = \frac{N_{correct\_first}}{N_{first}} \tag{8}$$

$$precision = \frac{N_{correct}}{N_{label}} \tag{9}$$

$$recall = \frac{N_{correct}}{N_{ground\_truth}} \tag{10}$$

where N_{first} is the number of samples with concept S in the first position in GT, $N_{correst\_first}$ is the number of the samples whose first concept in SCS is the same as that in GT, i.e. S, $N_{correct}$ is the number of samples having a given concept S in its SCS correctly, N_{label} is the number of samples having that concept in SCS, N_{ground_truth} is the number of samples having that concept in GT.

Table 2 shows the mean *S1_first* (MF), the mean precision (MP) and the mean recall (MR) over all concepts in SCS. KNN consistently outperforms NB, which conforms to the conclusion drawn by Benítez [6]. It also indicates that all metrics of SCCM are the biggest among the three methods, especially MR and MF of SCCM are much bigger than those of NB and KNN methods. So we have two reasons to expect that the annotation performance obtained by SCCM could be the best among the three methods. First, the larger the recall is, the more the SCS covers the correct concepts. Second, the first concept in SCS is the first evidence during Bayesian inference and the accuracy of the evidence is very crucial to the annotating results.

Table 2. The MP, MR and MF of semantic candidate set before inference

Method	MP	MR	MF	# concepts with recall>0	concepts with recall=0
NB	0.274	0.444	0.137	13	waterfall
KNN	**0.378**	0.490	0.182	12	waterfall , bridge
SCCM	**0.378**	**0.779**	**0.392**	**14**	\

After obtaining the SCS and the first concept of a testing frame, the system obtains the other concepts coexisting in the same frame from SCS by Bayesian Inference detailed in section 4. We also measure the performance of annotating by formula (9) and (10), where $N_{correct}$ is the number of the correct concepts that the system automatically annotated for a testing frame, N_{label} is the total number of the concepts that the system automatically annotates for a testing frame, N_{ground_truth} is the number of the actual concepts of a testing frame in GT.

Table 3 shows the average results of annotating using different methods of extracting SCS and two methods of constructing BN structure. First, it indicates that the inference performance on SCS obtained using SCCM is significantly higher than that using NB and KNN algorithm, no matter which method of constructing BN is used. Second, it shows that the recall values of all concepts in SCCM are larger than 0, and that of some concepts with low presence frequency in NB and KNN are zero. This is because KNN and NB algorithm are sensitive to the distribution of each concept in the training set. SCCM has not such problem because it is not sensitive to the presence frequency of each concept. Even a concept with low presence frequency can be assigned to a video shot only if its semantic class centre is close to the shot according to the visual features. It indicates that SCCM is more robust than NB algorithm and KNN algorithm. Third, IDABM performs a little better than TDABM, no matter which method of obtaining SCS is used.

Table 3. The result of automatic annotation. # con is the number of concepts with recall>0.

Method	NB			KNN			SCCM		
	MP	MR	#con	MP	MR	#con	MP	MR	#con
TDABM	0.277	0.414	13	0.355	0.47	11	0.406	0.731	14
IDABM	0.279	0.444	13	0.360	0.489	11	0.410	0.765	14

Figure 2 shows the automatic annotation of several testing samples using three methods to obtain SCS and two methods to construct BN. It is obvious that there is a main concept in each frame. For example, the main concept in Figure 2(a) is 'waterfall', that in Figure 2(b) is 'car' and that in Figure 2(c) is 'bridge'. All the above concepts have lower presence frequency in training set. SCCM can correctly capture these concepts, but NB and KNN algorithm can not. In most cases, SCCM is able to correctly annotate the main concepts, even though it annotates incorrectly with some concepts such as 'building' in Figure 2(a) and 'water' in Figure 2(b).

In general, SCCM algorithm outperforms significantly NB algorithm and KNN algorithm in annotating video shots. And IDABM performs a little better than TDABM without any prior knowledge.

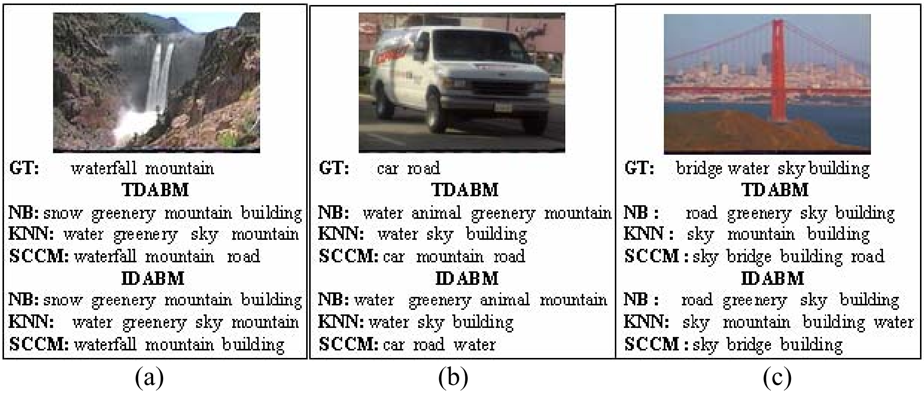

GT: waterfall mountain
TDABM
NB: snow greenery mountain building
KNN: water greenery sky mountain
SCCM: waterfall mountain road
IDABM
NB: snow greenery mountain building
KNN: water greenery sky mountain
SCCM: waterfall mountain building
(a)
GT: car road
TDABM
NB: water animal greenery mountain
KNN: water sky building
SCCM: car mountain road
IDABM
NB: water greenery animal mountain
KNN: water sky building
SCCM: car road water
(b)
GT: bridge water sky building
TDABM
NB : road greenery sky building
KNN : sky mountain building
SCCM : sky bridge building road
IDABM
NB : road greenery sky building
KNN: sky mountain building water
SCCM : sky bridge building
(c)

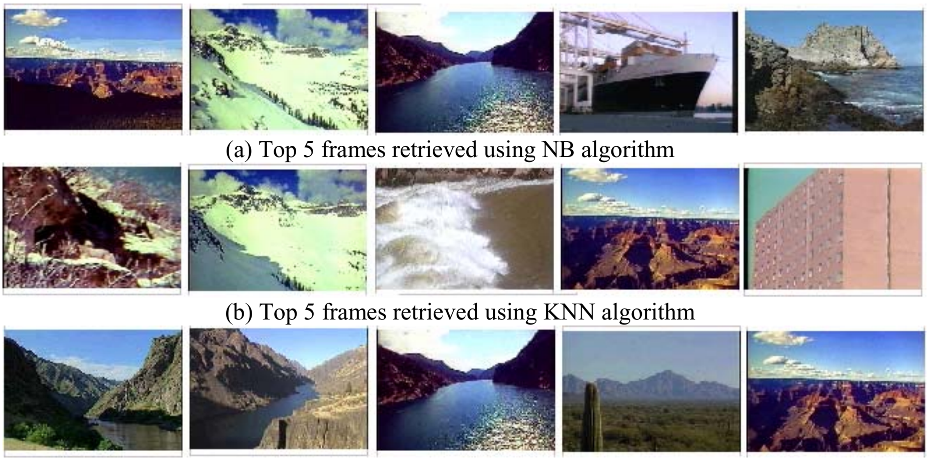
Fig. 2. The examples of automatic annotation of several methods

6.2 Results: Ranked Retrieval of Videos

In this section we turn our attention to the problem of ranked retrieval of key frames. In the retrieval setting we are given a text query Q={w_1, w_2, …, w_k} and a testing collection of unannotated key frames. For each testing frame *f*, Equation (7) is used to get the conditional probability $P(Q|f)$. All frames in the collection are ranked according to the conditional likelihood $P(Q|f)$. In our retrieval experiments, we use four sets of queries, constructed from all 1-, 2-, 3- and 4-word combinations of concepts that occur at least twice in the testing set. A frame is considered relevant to a given query if its manual annotation (ground-truth) contains all of the query words. We use precision and recall averaged over the entire query set as our evaluation metrics.

Table 4. The experiment result of retrieval

#concepts	Metric	NB		KNN		SCCM	
		Precision	Recall	Precision	Recall	Precision	Recall
1-concept	TDABM	0.277	0.414	0.361	0.471	0.410	0.731
	IDABM	0.279	0.444	0.364	0.491	**0.416**	**0.765**
2-concept	TDABM	0.168	0.313	0.194	0.332	0.261	0.435
	IDABM	0.176	0.341	0.196	0.355	**0.269**	**0.475**
3-concept	TDABM	0.117	0.154	0.178	**0.289**	0.258	0.214
	IDABM	0.130	0.205	0.195	**0.409**	**0.284**	0.318
4-concept	TDABM	0.111	0.125	0.139	0.313	0.25	0.313
	IDABM	0.111	0.125	0.139	0.313	**0.436**	**0.613**



(a) Top 5 frames retrieved using NB algorithm

(b) Top 5 frames retrieved using KNN algorithm

(c) Top 5 frames retrieved using SCCM algorithm

Fig. 3. Example of top 5 frames retrieved in response to the text query “mountain sky”. All the results of IDABM are the same with those of TDABM.

Table 4 shows the performance of our method on the four query sets, contrasted with performance of NB algorithm and KNN algorithm on the same data. We observe that our method substantially outperforms the NB algorithm and KNN algorithm on every query set except for the recall of SCCM compared with that of KNN in 3-concept query. It could be said SCCM has the same performance with KNN algorithm in this case because the sum of precision and recall in both methods are about equal. Figure 3 shows top 5 frames retrieved in response to the text query “mountain sky”.

We do not compare the results of our method with that of other papers because it is unfair to make a direct quantitative comparison with their method using different data set. It is a pity that there is no free and standard video data set to measure the algorithm performance now.

7 Conclusion

We have proposed a new approach to automatically annotate a video shot with a varied number of semantic concepts and to retrieve videos based on text queries. There are three main contributions of this work. The first one is to propose a simple but efficient method to automatically extract the semantic candidate set based on visual features. The second one is to propose IDABM algorithm to build the semantic network. IDABM has two advantages over TDABM. One is no need for users to provide the node ordering. The other is to reduce the time complexity from O(n^4) to O(n^2) when orienting the edges. The third one is to obtain the annotation with varied length from SCS by Bayesian Inference. Experiments show that SCCM is efficient and significantly outperformed KNN and NB method in obtaining SCS, though it is simple, and IDABM is better than TDABM algorithm in automatic semantic annotation.

References

[1] John R. Smith, Murray Campbell, Milind Naphade, Apostol Natsev, Jelena Tesic : Learning and Classification of Semantic Concepts in Broadcast Video. International conference on intelligence analysis (2005).

[2] Yan rong : Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval. Dissertation of Carnegie Mellon University (2005).

[3] Barnard, K., P. Duygulu, and D. Forsyth, N. de Freitas, D. Blei, and M.I. Jordan : Matching Words and Pictures. Journal of Machine Learning Research (JMLR), Special Issue on Text and Images Vol. 3. (2003) 1107-1135.

[4] Tseng, B.T., C.-Y. Lin, M.R. Naphade, A. Natsev, and J.R. Smith : Normalized Classifier Fusion for Semantic Visual Concept Detection. In Proc. of Int. Conf. on Image Processing (ICIP-2003), Barcelona, Spain (2003) 14-17.

[5] Milind Ramesh Naphade : A Probabilistic Framework For Mapping Audio-visual Features to High-Level Semantics in Terms of Concepts and Context. Dissertation of the University of Illinois at Urbana-Champaign (2001)

[6] Ana Belén Benítez Jiménez : Multimedia Knowledge: Discovery, Classification, Browsing, and Retrieval. Dissertation of Columbia University (2005)

[7] S. L. Feng, R. Manmatha and V. Lavrenko. Multiple Bernoulli Relevance Models for Image and Video Annotation. In CVPR, 2004

[8] Cheng, J., Greiner, R., Kelly,J. & Bell, D., Liu,w. Learning Belief Networks from Data: An Information Theory Based Approach. Artificial Intelligence. (2002)137(1-2):43-90.

[9] Cecil Huang: Inference in Belief Networks: A Procedural Guide. International Journal of Approximate Reasoning Vol. 11 New York (1994) 1-158.

Application of Novel Chaotic Neural Networks to Text Classification Based on PCA

Jin Zhang[1,3], Guang Li[2,*], and Walter J. Freeman[4]

[1] Department of Biomedical Engineering, Zhejiang University, Hangzhou, 310027, China
[2] National Laboratory of Industrial Control Technology, Zhejiang University,
Hangzhou, 310027, China
guangli@cbeis.zju.edu.cn
[3] Software College, Human University, Changsha, 410082, China
[4] Division of Neurobiology, University of California at Berkeley, Donner 101,
Berkeley, CA, 94720-3206, USA

Abstract. To model mammalian olfactory neural systems, a chaotic neural network entitled K-set has been constructed. This neural network with non-convergent "chaotic" dynamics simulates biological pattern recognition. This paper reports the characteristics of the KIII set and applies it to text classification. Compared with conventional pattern recognition algorithms, its accuracy and efficiency are demonstrated in this report on an application to text classification.

Keywords: Chaotic neural network, text classification, principal component analysis.

1 Introduction

Text classification [1] is the problem of automatically assigning predefined categories to natural language texts, based on their contents. Text classification applications are emerging as an important class of text processing applications, including indexing texts to support document retrieval, extracting information from texts, aiding human indexers in their tasks, and knowledge management using document classification agents.

Due to the increasing volume of information needing to be processed, many algorithms have been proposed in order to improve the performance of classification, such as support vector machine (SVM) [2], Bayes algorithm [3], and so on. Artificial neural network [4] is also widely used in text classification because of its classification and powerful nonlinear mapping ability.

Although traditional artificial neural networks simulate some primary features such as the threshold behavior and plasticity of synapses, their structures are still much simpler in comparison with biological neural system in real life. Based on the experimental study of the olfactory systems of rabbits and salamanders, a chaotic neural network mimicking the olfactory system called KIII model has been established. Built according to the architecture of the olfactory neural system, this

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1041 – 1048, 2006.

chaotic neural network can simulate EEG waveform observed in biological experiments well. Contrasting to conventional linear methods, the KIII model has strong nonlinear characteristics.

In this paper, we use the novel chaotic neural network mimicking olfactory system as a pattern classifier for text classification. Before classification, the features are extracted based on principal component analysis (PCA). The experimental results show that the KIII model has potential for text classification and other pattern classification tasks.

2 A Neural Chaotic Neural Network — The KIII Model

2.1 KIII Model

KIII model architecture follows olfactory system [5, 6], which consists of four main parts (shown in the left part of Fig.1). A schematic diagram of the KIII model, in which K0, KI and KII are included, is shown in the right part of Fig. 1. In K-set model, each node represents a neural population or cell ensemble. The dynamical behavior of each ensemble of the olfactory system is governed by equation (1).

$$\frac{1}{a \cdot b}\left[x_i''(t) + (a+b)x_i'(t) + a \cdot b \cdot x_i(t)\right] = \sum_{j \neq i}^{N}\left[W_{ij} \cdot Q(x_j(t), q_j)\right] + I_i(t) \qquad (1)$$

$$Q(x_i(t), q) = \begin{cases} q(1 - e^{-(\exp(x(t))-1)/q}) & x(t) > x_0 \\ -1 & x(t) < x_0 \end{cases} \qquad (2)$$

$$x_0 = \ln(1 - q\ln(1 + 1/q))$$

Here i=*1…N*, where *N* is the number of channels. In equation (1), $x_i(t)$ represents the state variable of *i*th neural population, $x_j(t)$ represents the state variable of *j*th neural population, which is connected to the *i*th, while W_{ij} indicates the connection strength between them. $I_i(t)$ is an input function, which stands for external stimulus. Physiological experiments confirm that a linear second-order derivative is an appropriate choice. The parameter a = 0.220msec^{-1}, b = 0.720 msec^{-1} reflect two rate constant derived from the electro-physiological experiments. $Q(x_j(t),q_j)$ is a nonlinear sigmoid function derived from Hodgkin-Huxley equation and is expressed as equation (2). In the equation (2), q represents the maximum asymptote of the sigmoid function, which is also obtained from biological experiments. In different layers, q is different, for example, q is equal to 5 in OB layer and equal to 1.824 in PG layer. And x_0 can reflect the state of neuron should be excitatory or inhibitory.

The KIII model describes the whole olfactory neural system. It includes populations of neurons, local synaptic connection, and long forward and distributed time-delayed feedback loops. In the topology of the KIII network, *R* represents the olfactory receptor, which is sensitive to the odor molecules, and provides the input to the KIII network. The PG layer and OB layer are distributed, which contains *n* channels. The AON and PC layer are only composed of single KII network. The parameters in the KIII network were optimized by measuring the olfactory evoked

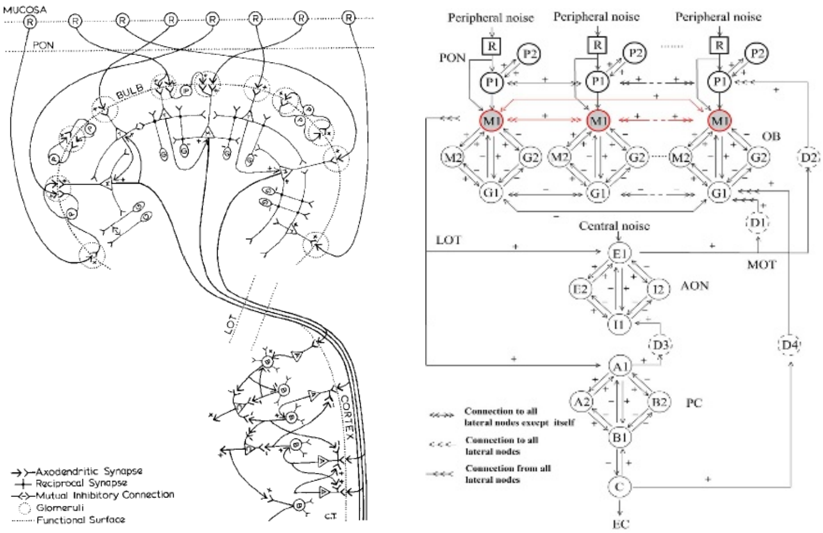


Fig. 1. Topological diagram for the neural system (left) and KIII network (right)

potentials and EEG, simulating their waveforms and statistical properties, and fitting the simulated functions to the data by means of nonlinear regression. After the parameter optimization, the KIII network generates EEG-like waveform with $1/f$ power spectra [7].

Compared with well-known low-dimensional deterministic chaotic systems such as the Lorenz or Logistics attractors, nervous systems are high-dimensional, non-autonomous and noisy systems [8]. In the KIII network, independent rectified Gaussian noise was introduced to every olfactory receptor to model the peripheral excitatory noise and single channel of Gaussian noise with excitatory bias to model the central biological noise sources. The additive noise eliminated numerical instability of the KIII model, and made the system trajectory stable and reliable under statistical measures, which meant that under perturbation of the initial conditions or parameters, the system trajectories were robustly stable [9]. Because of this stochastic chaos, the KIII network not only simulated the chaotic EEG waveforms, but also acquired the capability for pattern recognition, which simulated an aspect of the biological intelligence, as demonstrated by previous applications of the KIII network to recognition of one-dimensional sequences, industrial data and spatiotemporal EEG patterns [10], which deterministic chaos can't do, owing to its infinite sensitivity to initial conditions.

2.2 Learning Rule

We study the n-channel KIII model, which means that the R, P and OB layer all have n units, either K0 (R, P) or KII (OB). When stimulus is presented to some of the n channels while other channels receive the background input (zero), the OB units in these channels experience a state transition from the basal activity to a limit cycle attractor and the oscillation in these units increases dramatically. Therefore, we choose the activity of the M1 node in the OB unit to be the scale to indicate the extent to which the channel is excited. In other words, the output of the KIII set is taken as a $1 \times n$ vector at the mitral level to express the AM patterns of the local basins, specifically the input-induced oscillations in the gamma band (20 Hz to 80 Hz) across the n channels during the input-maintained state transition. To put it clearly, the activity of M1 node is calculated as the root mean square (RMS) value of the M1 signal over the period of stimulation.

There are two kind of learning rules [11]: Hebbian and habituation learning. The rule of Hebbian learning reinforces the desired stimulus patterns while habituation learning decreases the impact of the background noise and the stimuli that are not relevant or significant. According to the modified Hebbian learning rule, when two nodes become excited at the same time with reinforcement, their lateral connection weight is strengthened. On the contrary, without reinforcement the lateral connection is weakened.

Let $P_M(i)$ donate the activity of the *i*th M1 node in the OB layer, P_M indicate the average value of all the $P_M(i)$ ($i = 1,\ldots, n$), $W_{M(i)\text{->}M(j)}$ represent the connection weight from the *i*th M1 node to the *j*th M1 node, h_{Heb} and h_{Hab} denotes the connection weight when two nodes are excited or not separately.

The algorithm can be described as follows:

```
For $i = 1...N$
  For $j = 1...N$
        IF $P_M(i)>(1+K)*P_M$ AND $P_M(j)>(1+K)*P_M$ AND $i != j$ Then
              $W`_{M(i)\text{->}M(j)} = h_{Heb}$ $(i = 1,\ldots,N)$
        ElseIF $i == j$ Then
              $W`_{M(i)\text{->}M(j)} = 0$
        Else
              $W`_{M(i)\text{->}M(j)} = h_{Hab}$ $(i = 1,\ldots,N)$
        End
  End      //loop for j
End      //loop for i
```

The bias coefficient $K>0$ is introduced in order to avoid the saturation of the weight space. The habituation constitutes an adaptive filter to reduce the impact of environmental noise that is continuous and uninformative. In our simulation, the habituation exists at the synapse of the M1 nodes on other nodes in the OB and the lateral connection within the M1 layer. It is implemented by incremental weight decay (multiply with a coefficient $h_{hab} <1$) of all these parameters at each time step continuously through the entire learning period.

3 Application of KIII Model to Text Classification

Text classification is defined as follows: given a natural language input text and a set of predefined categories, determine which categories are most similar to the input text. In order to reduce the dimension of features, PCA (Principal Component Analysis) is used to extract the features of text and how KIII model is used to classify text is shown in this section.

3.1 The Process of Text Classification

The whole process of text classification is shown in Fig.2. The whole process includes several steps. First, all texts are transformed term-document matrix (T-D matrix). In the T-D matrix, the column denotes the terms used in the text dataset and each row denotes one text. Each value in one row is the frequency of terms used in one text. Second, T-D matrix is transformed based on PCA to reduce its dimensionality. Third, the transformed T-D matrix is divided into two parts: training set and testing set. Generally, testing set is about one third of the whole dataset. Then, the training set is used to train KIII model to learn new patterns. Finally, testing set is input to KIII model and classified to different category.

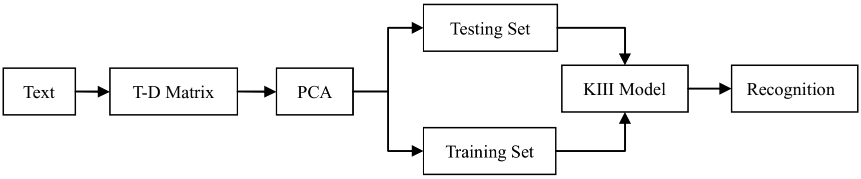


Fig. 2. The process of text classification

3.2 Feature Extraction

A problem in text classification is how to reduce the high dimensionality of feature space, which normally consists of tens or even hundreds of thousands of unique words, phrases, and perhaps other semantic entities that occur in the natural language texts to be categorized. Hence, reducing the dimensionality is of great importance for neural networks to be applied to text categorization.

PCA [12] is a dimensionality reduction method that focuses on finding the basis vectors which best represent the data in a minimum squared error sense. Consider a p -dimensional input vector $x = (x_1, x_2, \dots, x_p)$, according to correlation among p attributes, PCA use k ($k<p$) principal components to express p attributes completely. K principal components, which replace raw data, are a given linear combination of raw p attributes and can be obtained by calculating eigenvector.

PCA solution is according to equation (3) and (4). In equation (3), X is a matrix of size $M\times N$ and m is the mean value. PCA involves solving the eigenvalue problem. In equation (4), the covariance matrix C is symmetric and positive semi-definite. Therefore, the eigenvectors computed in equation (4) form an orthogonal basis that best represents the variance of the training data in the minimum mean squared error sense. In this paper, we use PCA to transform the raw data and remove separately those attributes whose contribution is less than 0.0015, 0.001 and 0.0007. After that, each text can be denoted by 108, 164 and 229 dimension feature.

$$C = \sum_{1}^{N} (x_i - m)(x_i - m)^T = XX^T \tag{3}$$

$$XX^T v = Cv = \lambda v \tag{4}$$

3.3 Performance Evaluation

3.3.1 Dataset Description

In this paper, we select 20-newsgroups corpus [13] to evaluate the performance of KIII model. 20-newsgroups corpus is a popular data set when evaluating the performance of model. It is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. One thousand Usenet articles were taken from each of the following 20 newsgroups. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware/ comp.sys.mac.hardware), while others are highly unrelated (e.g misc.forsale/soc. religion.christian).

3.3.2 Experiment Results

In the simulation, 400 articles are taken from each class as testing set and others are taken as training set. The experiment results are listed in Table.1. In Table.1, Class *i* (*i=1…20*) denotes separately alt.atheism, comp.sys.ibm.pc.hardware, misc.forsale, rec.motorcycles, sci.space, talk.politics.guns, rec.sport.baseball, talk.religion.misc, comp.windows.x, comp.os.ms-windows.misc, comp.graphics, rec.autos, talk.politics. misc, sci.electronics, sci.med, comp.sys.mac.hardware, rec.sport.hockey, talk.politics. mideast, sci.crypt, soc.religion.christian.

Table 1. The simulation result of KIII model

	50			100		
	108	164	229	108	164	229
Class 1	0.675	0.685	0.663	0.655	0.680	0.680
Class 2	0.895	0.893	0.888	0.895	0.888	0.873
Class 3	0.768	0.773	0.775	0.813	0.803	0.803
Class 4	0.768	0.803	0.803	0.785	0.798	0.818
Class 5	0.728	0.733	0.753	0.698	0.730	0.760
Class 6	0.670	0.673	0.638	0.673	0.680	0.670
Class 7	0.815	0.800	0.805	0.838	0.833	0.833
Class 8	0.660	0.643	0.650	0.675	0.703	0.693
Class 9	0.693	0.695	0.663	0.683	0.703	0.683
Class 10	0.715	0.713	0.715	0.753	0.753	0.765
Class 11	0.725	0.730	0.740	0.820	0.830	0.825
Class 12	0.785	0.785	0.775	0.75	0.753	0.750
Class 13	0.690	0.668	0.660	0.723	0.723	0.720
Class 14	0.788	0.783	0.770	0.803	0.795	0.790
Class 15	0.728	0.758	0.753	0.845	0.828	0.820
Class 16	0.855	0.855	0.863	0.888	0.880	0.878
Class 17	0.760	0.748	0.738	0.830	0.795	0.793
Class 18	0.663	0.660	0.655	0.693	0.708	0.700
Class 19	0.760	0.785	0.768	0.818	0.840	0.845
Class 20	0.538	0.535	0.518	0.775	0.780	0.768
Ave.	**0.734**	**0.736**	**0.730**	**0.771**	**0.775**	**0.774**

In Table.1, the first row (50 or 100) denotes the number of article in training set. The second row (108, 164 and 229) denotes the number of features extracting from one article. Each article is used only once during training. It can be seen that KIII model needs few training times to reach steady state. For example, only 50 articles and 50 training times is enough to KIII model which can reach over 70% accuracy. Table.2 denotes the performance of BP neural network. The number of training time is equal. It is shown that BP network has low accuracy, less than 15%, at the same training conditions, even more samples are used to train BP neural networks.

Table 2. The simulation results of BP

		The number of training articles						
		50	100	200	300	400	500	600
The number of feature	108	0.096	0.097	0.086	0.120	0.134	0.083	0.096
	164	0.111	0.075	0.107	0.112	0.117	0.101	0.153
	229	0.069	0.112	0.085	0.137	0.117	0.130	0.122

4 Conclusion

KIII network is a kind of nonconvergent "chaotic" neural network, which is derived directly from the biological neural system. Its mechanism for pattern recognition is totally different from other artificial neural networks (ANN). As a result, it not only can simulate the experimental EEG waveform, but also has the ability to recognize complex patterns, such as text classification. Taking after the biological olfactory neural system being good at learning new odors, KIII model requires only a few learning trials to set up the basins of attraction of the memory patterns with robust tolerance for noise and speech variability.

Mimicking biological neural systems is shown to be an efficient way to handle complicated pattern recognition problems. But there remains much work to be done, including the parameter selection, the training algorithms and so on. All of them are included in our further works.

References

1. Wai Lam, M. Ruiz, P. Srinivasan.: Automatic text categorization and its application to text retrieval. IEEE Transactions on Knowledge and Data Engineering. 11(1999) 865–879
2. Jian-Guo Zhou, Kai Wang, Jing Wu.: A method of Chinese text categorization based on proximal support vector machine. In: Proc. of 2005 International Conference on Machine Learning and Cybernetics. 3(2005) 1615 – 1619
3. Fei Yu, Ji-yao An, Hong Li, Miao-liang Zhu, Ouyang Yang.: Intelligence text categorization based on Bayes algorithm. In: Proc. of International Conference on Information Acquisition. 1(2004) 347-350

4. P. Manomaisupat, K. Abmad.: Feature selection for text categorization using self-organizing map. In: Proc. of International Conference on Neural Networks and Brain (ICNN&B'05), 3(2005) 1875-1880
5. Huang-Jen Chang; Walter J. Freeman: Biologically modeled noise stabilizing neurodynamics for pattern recognition. In: Proc. of Int. J. Bifurcation and Chaos, 8(1998) 321-345
6. Yao, Y. & Freeman, W. J.: Pattern recognition in olfactory systems: modeling and simulation, In: Proc. of the 1989 International Joint Conference on Neural Networks, (1989) 699-704
7. Chang, H.J. & Freeman, W. J.: Optimization of olfactory model in software to give power spectra reveals numerical instabilities in solutions governed by aperiodic (chaotic) attractors. Neural Networks, 11(1998) 449-466
8. Freeman, W. J. & Kozma, R.: Biocomplexity: adaptive behavior in complex stochastic dynamic systems. Biosystems, 59(2001) 109-123
9. Chang, H.J. & Freeman, W. J. Local homeostasis stabilizes a model of the olfactory system globally in respect to perturbations by input during pattern classification. In: Proc. of Int. J. Bifurcation and Chaos, 8(1998) 2107-2123
10. Kozma, R. & Freeman, W. J.: Chaotic resonance – methods and applications for robust classification of noisy and variable patterns. In: Proc. of Int. J. Bifurcation and Chaos, 11(2001) 1607-1629
11. Guang Li, Jin Zhang, You Wang and Walter J. Freeman.: Face Recognition Using a Neural Network Simulating Olfactory Systems. Lecture Notes in Computer Science, 3972(2006) 93-97
12. I.T. Jolliffe: Principal component analysis. New York: Springer, 1986
13. http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

Lip Localization Based on Active Shape Model and Gaussian Mixture Model

Kyung Shik Jang[1], Soowhan Han[1], Imgeun Lee[2], and Young Woon Woo[1]

[1] Department of Multimedia Engineering, Dong-Eui University, San 24, Gaya-Dong, Pusanjin-Gu, Pusan, 614-714, Korea

[2] Department of Film and Visual Engineering, Dong-Eui University, San 24, Gaya-Dong, Pusanjin-Gu, Pusan, 614-714, Korea

ywwoo@deu.ac.kr

Abstract. This paper describes an efficient method for locating lip. Lip deformation is modeled by a statistically deformable model based on Active Shape Model(ASM). In ASM based methods, it is assumed that a training set forms a cluster in shape parameter space. However if there are some clusters in shape parameter space due to an incorrect position of landmark point, ASM may not be able to locate new examples accurately. In this paper, Gaussian mixture is used to characterize the distribution of shape parameter. The Expectation Maximization algorithm is used to determine the maximum likelihood parameters of Gaussian mixture. During search, we resolved the updated locations by projecting a shape into the shape parameter space by using Gaussian mixture. The experiment was performed on many images, and showed very encouraging result.

1 Introduction

Recently, there is an increasing requirement for a system to track and locate human lip[1][2]. Human lip has much more information than any other face features, so the lip information could be used in image coding[2]. To improve the performance of speech recognition, the lip information is used together with the acoustic signal[3][4]. The information is also be applied to the graphic animation systems, which need it for generating the lip shape of the speaker[2][4].

Accurately and robustly tracking lip motion in image sequences is especially difficult because lip are highly deformable and they vary in shape and color. Gradient based techniques[5][6] for edge detection of lip often fail due to the poor contrast between lip and surrounding skin region. For methods using color information to build a parametric deformable model for the lip contour, these require optimization technique to refine estimates of contour model to the human lip[7][8]. Many papers have described the applications of active contour for lip boundary detection[9][10]. The snake and active contour methods are able to resolve fine contour details but shape constraints are difficult to incorporate. Furthermore, the snake methods often converge to the wrong result when the lip edges are not distinct or when the lip color is very close to the face skin. Many

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1049–1058, 2006.

methods have localized only the outer lip contour when the mouth is closed because the presence of tongue and teeth could obscure the inner contour when the mouth is open. Delmas[9] and Lievin[10] have proposed a statistical approach based on Markov random field to segment lip using the color information and the motion in a spatiotemporal neighborhood. This method has shown good result at the outer lip contour, but shown bad result at the inner lip contour. Methods[4][11] using active shape model(ASM) have shown good result in localizing not only the outer lip contour but also the inner one. However they have shown poor results in some cases. In general ASM based methods[12][13], it is assumed that the training set forms one cluster in the parameter space. However it is not true in all case. If the distribution of shape parameter has some peaks, a value between two peaks is meaningless and likely to generate an implausible shape. For example, because the landmark points are placed along the lip contour by hand labeling and the lip boundary is not clear, it is likely for the position of the landmark point to be incorrect. In such cases, it may be some peaks in shape parameter space and the shape model using one Gaussian model does not represent the training data accurately.

This paper describes an efficient method for locating lips. The lip shape is represented as a set of landmark points and lip deformation is modeled by a statistically deformable model based on ASM. Gaussian mixture model(GMM) is used to characterize the distribution of a shape parameter because the distribution may not be uniform in shape. The Expectation Maximization algorithm is used to determine the maximum likelihood parameters of Gaussian mixture. During search, we resolve the updated locations by projecting a shape into the shape parameter space by using GMM. The experiment has been performed on the images from Tulips 1 database, and excellent results have been obtained.

2 Lip Model

2.1 Lip Shape Model

The lip shape is described by a set of 41 landmark points as shown in Fig. 1(a). The lip shape of ith training image is described by a shape vector containing the coordinates of the landmark points as shown in equation (1). The lip shape model is represented using equation (2), where $\overline{\mathbf{X}}$ is a mean shape, P is a matrix of the first t column eigenvectors corresponding to the largest eigenvalues and a shape parameter b is a vector containing the weights for each eigenvector[4].

$$\mathbf{X}_i = (x_{1i}, y_{1i}, x_{2i}, y_{2i}, \cdots, x_{41i}, y_{41i})^T \tag{1}$$

$$\begin{aligned}&\mathbf{X} = \overline{\mathbf{X}} + \mathbf{P}\,\mathbf{b} \\ &\quad \textit{where},\ \mathbf{P} = [\mathbf{P}_1\ \mathbf{P}_2\ \mathbf{P}_3 \cdots \mathbf{P}_t], \quad \lambda_i \geq \lambda_{i+1}\ , \quad \mathbf{b} = (b_1, b_2, \cdots, b_t)^T\end{aligned} \tag{2}$$

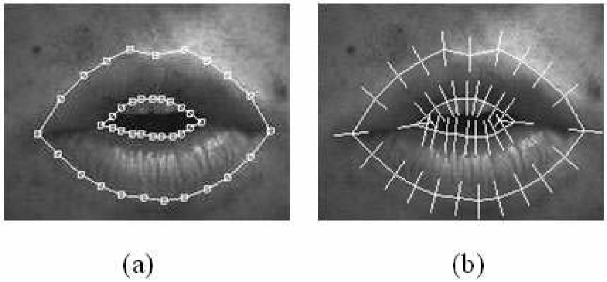


Fig. 1. Lip shape model and intensity profile

2.2 Lip Boundary Model

Lip boundary model is constructed by statistical analysis of the appearance of the image intensity in the neighborhood of each landmark point. As shown in Fig. 1 (b), for every landmark point j in the image i of the training set, we choose to sample one dimensional profile g_{ij} of length N_p perpendicular to the contour and centered at the point j as shown in equation (3). It is sensitive to the light condition to use the absolute gray level value, so we normalize the vector g_{ij} to obtain g^n_{ij} using equation (4). A mean profile $\bar{g}^n_{ij}$ and a covariance matrix S_j are derived. This is repeated for all landmark points at outer and inner contour, giving a profile model for each point.

$$\mathbf{g}_{ij} = (g_{ij1}, g_{ij2}, g_{ij3}, \cdots, g_{ijNp})^T \quad \textit{where}, \quad g_{ijk} \textit{ is a pixel value.} \tag{3}$$

$$\mathbf{g}^n_{ij} = \frac{\mathbf{g}'_{ij}}{\sum \left| g'_{ijk} \right|} \qquad \textit{where},\ g'_{ij} = \{g_{ijk} | g_{ij(k+2)} - g_{ijk},\ k = 1, \cdots, N_p - 2\} \tag{4}$$

During searching a lip, we sample an image feature profile H_j of length $M_p(M_p > N_p)$ either side of a point j of the lip contour produced by the shape model. The normalized profile H^n_j is derived by using equation (4). We make h^n_j by selecting a sample of length N_p at each of the $M_p - N_p + 1$ possible positions along H^n_j and compare it with $\bar{g}^n_j$ which is the profile model at jth landmark point. The updated location for current point is selected by choosing a point that minimizes a function in equation (5).

$$D(\mathbf{h}^n_j) = (\mathbf{h}^n_j - \overline{\mathbf{g}^n_j})^T \mathbf{S}^{-1}_j (\mathbf{h}^n_j - \overline{\mathbf{g}^n_j}) \tag{5}$$

3 Gaussian Mixture for Shape Parameters

In equation (2), the lip shape model is a function of a shape parameter b. In general ASM based methods, it is assumed that a training set forms one cluster

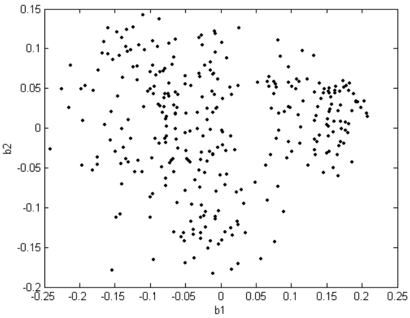


Fig. 2. The distribution of a shape parameter

in the parameter space. However, because the landmark points are placed along the lip contour by hand labeling and the lip boundary is not clear, it is likely for the position of the landmark points to be incorrect. In such cases, it may be some peaks in shape parameter space, a shape parameter value between two peaks is meaningless and likely to generate an implausible shape. Therefore, the shape model does not represent the training data accurately. Fig. 2 shows the distribution of $b_1$and b_2, which are selected from the a shape parameter b. It is desirable that the distribution be represented by Gaussian mixture rather than by single Gaussian.

In this paper, the distribution of a shape parameter is represented with Gaussian mixture[14]. With k mixture components, Gaussian mixture is represented in equation (6), where x is a feature vector, a parameter vector θ_i become to $\{\mu_i,\ \sum_i\}_{i=1}^{k}$. $P(i)$ is the mixture parameter representing a prior probability with which ith mixture component occurs. $g(x|\theta_i)$ represents a multi-variate Gaussian density function which is parameterized by θ_i.

$$\begin{aligned} G(\mathbf{x}|\Theta) &= \sum_{i=1}^{k} p(i) g(\mathbf{x}|\theta_i) \\ \mathit{where},\quad & \sum_{i=1}^{k} p(i) = 1,\ p(i) \geq 0, \quad \Theta\ =\ \{p(i),\ \theta_{\mathrm{i}}\}_{i=1}^{k} \end{aligned} \tag{6}$$

The Expectation-Maximization(EM) algorithm[14] is used to find maximum likelihood parameter estimates for Gaussian mixture. The EM algorithm consists of E-step and M-step. With given parameter values, E-step calculates an average likelihood function by using equation (7). M-step does maximum likelihood parameter estimates for data by using equation (8). The process is iterated until a likelihood function in equation (9) converges, where n is the total number of data.

$$P_i(x) = \frac{p(i)g(x|\theta_i)}{\sum\limits_{i=1}^{k} p(i)g(x|\theta_i)} \tag{7}$$

$$p(i) = \frac{\sum_{j=1}^{n} P_i(x_j)}{n}, \quad \mu_i = \frac{\sum_{j=1}^{n} P_i(x_j)x_j}{\sum_{j=1}^{n} P_i(x_j)}$$
$$\Sigma_i = \frac{\sum_{j=1}^{n} P_i(x_j)(x_j-\mu_i)(x_j-\mu_i)^T}{\sum_{j=1}^{n} P_i(x_j)} \tag{8}$$

$$L(\Theta) = \sum_{j=1}^{n} \log G(x_j|\Theta) \tag{9}$$

4 Lip Localization

4.1 Initial Position for Search

To reduce the dependency of initial position, we find out a line connecting two corner points of lip. The y-coordinate of the line is used as y-coordinate for the initial search. For each column of the image, y-coordinate of the smallest pixel value is found by using equation (10), where $I(x, y)$ represents a pixel value at coordinate (x, y). H and W represent the height and the width of the image, respectively. K_1 and K_2 are constants for all images. The highest peak of $L(y)$ in equation (11) gives the y-coordinate of the line connecting two corner points of lip. $T(x)$ varies from 0 to 1. $L(y)$ for a sample image is shown in Fig. 3(a) and the result is shown in Fig. 3(b). In Fig. 3(a), y-axis is an image height and x-axis is $L(y)$.

$$M(x) = \arg\min_{y \in H} I(x, y) \tag{10}$$

$$L(y) = \sum_{x \in W} T(x) \quad \textit{where},\ T(x) = \frac{K_1 - \cosh\left(\frac{M(x)-\frac{H}{2}}{\frac{H}{K_2}}\right)}{K_1} \tag{11}$$

4.2 Lip Localization Using Gaussian Mixture Model

Using lip shape model, a lip contour is generated. A region of an image in a neighborhood of each landmark point is examined. The best matching point to which the landmark point is moved is a point with the smallest Mahalanobis distance(equation (5)) between the model profile and the image feature profile. Each point is moved independently, thus it deforms the lip shape to implausible one which is a new lip contour. The deformed shape is corrected by rescaling a shape parameter using GMM. The iterative search process is described below. θ, t and s are rotation, translation (t_x, t_y) and scale factor respectively.

(1) Generate a lip contour from the lip shape model in equation (2) using a shape parameter b, and perform linear transform using equation (12).

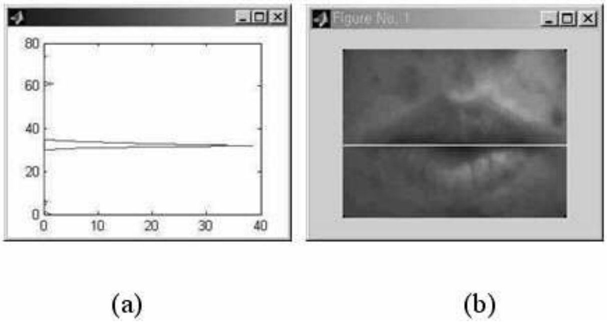


Fig. 3. A line connecting the both corner points of a lip

(2) Examine the image region in a neighborhood of each landmark point, and find out new contour Y.
(3) Obtain (θ, t, s) using equation (13). Calculate y using $\mathbf{y} = T^{-1}_{(\theta,t,s)}(\mathbf{Y})$.
(4) find out a new shape parameter b using equation (14), and then split it into two parts, $b_a = (b_1, \ldots, b_k)^T$ and $b_b = (b_{k+1}, \ldots, b_t)^T$. Gaussian mixture model is used for b_a, and one Gaussian model is used for b_b.
(5) Using the Gaussian mixture $G(x)$, check $G(b_a) > P_t$. If it fails, the ith component of Gaussian mixture is selected using equation (15) and b_a is rescaled according to equations (16) and (17). P_t and $\mathrm{D}_{\max}$ are constants. Check the condition for b_b using equation (18), where λ_i is the ith eigenvalue in the lip shape model. If not satisfied, it is rescaled using equation (19).
(6) Iterate from step 1 to step 5 until b and (θ, t, s) converge.

$$T_{(\theta,t,s)}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \tag{12}$$

$$(\theta, t, s) = \arg\min |T_{(\theta,t,s)}(\overline{\mathbf{X}}) - \mathbf{Y}|^2 \tag{13}$$

$$\mathbf{b} \;=\; \mathbf{P}^T(\mathbf{y} - \bar{\mathbf{X}}) \tag{14}$$

$$i = \arg \max_{i \in \{1,2,\cdots,k\}} g(\mathbf{b}_a|\theta_i) \tag{15}$$

$$D^2_{ma} = (\mathbf{b}_a - \mu_i)^T \mathbf{\Sigma}^{-1}_i (\mathbf{b}_a - \mu_i) \tag{16}$$

$$\mathbf{b}_a = (\mathbf{b}_a - \mu_i)\frac{D_{\max}}{D_{ma}} + \mu_i \tag{17}$$

$$D^2_{mb} = \sum_{i=k}^{t}(\frac{b^2_{bi}}{\lambda_i}) \leq D^2_{\max} \tag{18}$$

$$\mathbf{b}_b = \mathbf{b}_b\frac{D_{\max}}{D_{mb}} \tag{19}$$

5 Experiments and Analysis

We used the images of the Tulips 1 database of isolated digits[15] for experiments. It consists of 96 gray image sequence of 12 speakers each saying the first four digits in English twice. We referred to the set of words spoken the first times as Set 1 and the set of words spoken the second times as Set 2. In this paper, the lip shape model was built using 300 images from Set 1. We used 12 shape modes t for the lip shape model. For experiment, 350 images in Set 2 were used. An initial value for parameter $(\theta,\ t,\ s)$ is $(0,\ (Y_{Lip},\ W/2),\ 70)$. Y_{Lip} represents y-coordinate of a line connecting two corner points of lip, and W is an image width. We have done the experiments using Gussian mixture with 2∼10 mixture components. As a result, Gaussian mixture with 4 mixture components was used for $b_1 \sim b_4$.

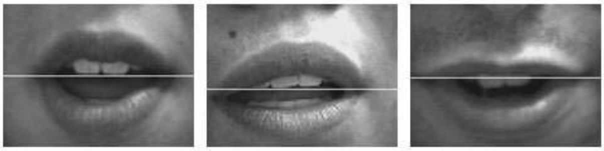


Fig. 4. Examples of initial position for searching lip

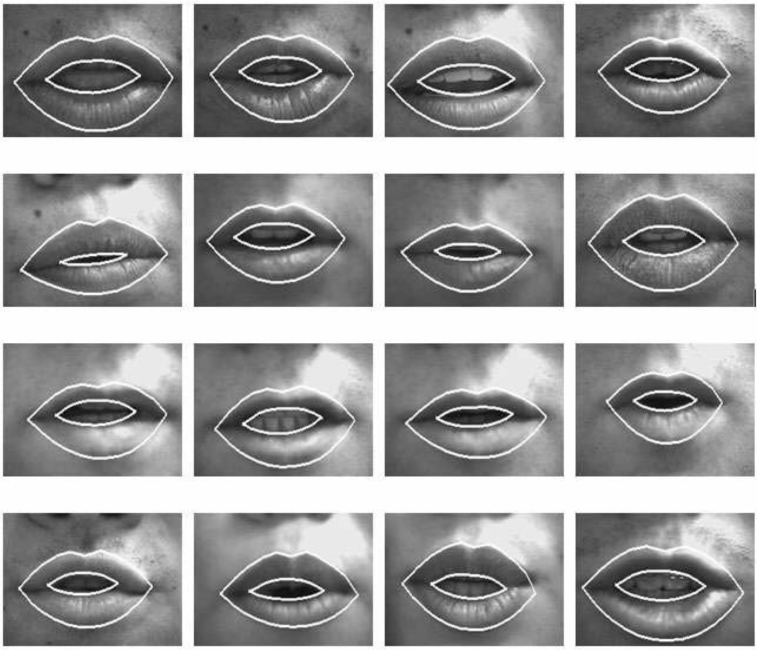


Fig. 5. Lip localization examples

Table 1. Average localization results

	Proposed method	ASM without GMM
Truth Positive %	81.15	76.94
False Negative %	18.85	23.06
True Negative %	95.65	95.15
False Positive %	4.35	4.85
Error %	8.57	10.31

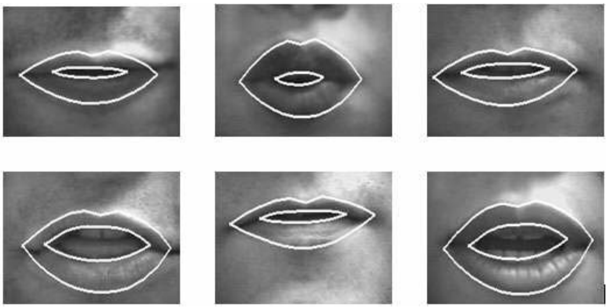


Fig. 6. Result by our method

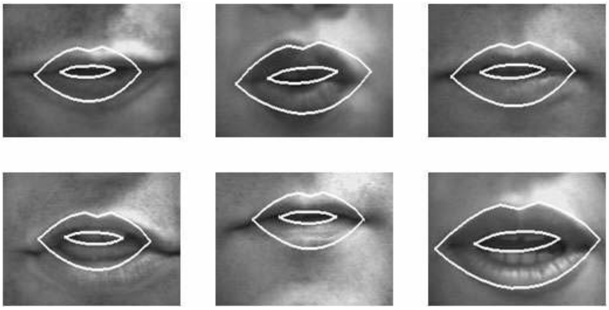


Fig. 7. Result by ASM without GMM

In Fig. 4, there are some examples of the initial positions for lip. Examples of lip localization for all subjects are shown in Fig. 5. To evaluate the proposed method, we implemented the proposed method(ASM with GMM) and ASM without GMM[4][13]. Table 1 shows the average lip localization results by our method along with results by ASM without GMM for all test images. All the test images were labeled by hand and were used as the ground truth. True-positive means that lip pixels are classified to lip pixels and false-negative means that lip pixels are classified to non-lip pixels. True-negative means that non-lip pixels are classified to non-lip pixels and false-positive means that non-lip pixels are classified to lip pixels.

Fig. 6 shows the result by our method and Fig. 7 shows the result by ASM without GMM. Because their shapes are not ordinary, they seem to form different cluster from ordinary shape in shape parameter space. In the previous methods using a single Gaussian, when projecting into shape space, it is possible for a shape parameter to be projected into local minima between some peaks, which would generate an implausible shape. However, in our method, a shape parameter would not be projected into local minima between some peaks because of GMM. For the outer lower lip, some errors occurred because there was no obvious gray level difference between the lips and their neighborhood skin. The errors in the inner lip contour occurred mainly due to the gradient originating from the teeth and tongue.

6 Conclusions

In this paper, we describe an efficient algorithm for lip localization based on ASM and GMM. The distribution of a shape parameter is modeled by Gaussian mixture. We have used the EM algorithm to determine the parameter. The proposed method was tested to many samples of various shapes and the result showed that it localizes correctly the lip shape that was not localized by ASM without GMM. The better performance may be obtained by improving the searching method with the profile information of the landmark points, and this is left to the next topic.

References

1. Mlrhosseinl A. R., H. Yan and K. M. Lam, "Adaptive Deformable Model for Mouth Boundary Detection", Optical Engineering, Vol. 37 No. 3(1998), pp. 869-875.
2. Oliver N., A. Pentland, "LAFTER: Lips and Face Real Time Tracker", Proceedings of the 1997 Conf. on Computer Vision and Pattern Recognition, (1997), pp. 123-129.
3. Kaucic R.. A. Blake, "Accurate, Real-Time, Unadorned Lip Tracking", Proceedings of the 6th International Conf. on Computer Vision, pp. 370-375, 1998.
4. Iain Matthnews, Timothy F. Cootes, J. Andrew Banghan, Stephen Cox and Richard Marvey, "Extractoin of Visual Features for Lipreading", IEEE Tans. on Pattern Recognition and Machine Analysis, Vol 24, No. 2, , pp. 198-213, Feb. 2002.
5. L. Zhang, "Estimation of the mouth features using deformable templates", IEEE International Conference on Image Processing, Vol. III, pp. 328-331, 1997.
6. M. Lievin, F. Luthon, "A Hierarchical Segmentation Algorithm for Face Analysis : Application to Lipreading", IEEE Conf. on Multimedia & Exposition'2000, August, 2000.
7. Wark T., Sridharan and V. Chandran, "An Approach to Statistical Lip Modelling for Speaker Identification via Chromatic Feature Extraction", Proceedings of the 14th International Conf. on Pattern Recognition, Vol. 1, pp. 123-125, 1998.
8. T. Wark, S. Sridharan, "A Syntatic Approach to Automatic Lip Feature Extraction for Speaker Identification", IEEE International Conference on Acoustics, Speech and Signal Processing, p 1227 - , 1998

9. Delmas P., Y. Coulon and V. Fristot, "Automatic Snakes for Robust Lip Boundaries Extraction", IEEE International Conf. on Acoustics, Speech and Signal Processing, Vol. 6, pp. 3069-3072, 1999.
10. Lievin M., F. Luthon, "Unsupervised Lip Segmentation under Natural Conditions", IEEE International Conf. on Acoustics, Speech and Signal Processing, Vol. 6, pp. 3065-3068, 1999.
11. Luettin, J, and Thacker, NA, "Speechreading using probabilistic models," Computer Vision and Image Understanding, vol. 65, pp. 163-178, 1997.
12. M. B. Stegmann, R. Fisker, "On Properties of Active Shape Models", Informatics and Mathematical Modelling, Technical University of Denmark, 2000.
13. Cootes, Taylor, Cooper and Graham, "Active Shape Models-Their Training and Application," Computer Vision and Image Understanding, Vol. 61, No. 1, pp. 38-59, 1995.
14. Chad Carson, Serge Belongie, Hayit Greenspan, Jitendra Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying", IEEE Tans. on Pattern Recognition and Machine Analysis, Vol. 24, No. 8, pp. 1026-1038, Aug. 2002.
15. Movellan J. R., "Visual Speech Recognition with Stochastic Networks", Advances in Neural Information Processing System. Vol. 7, MIT Press Cambridge, 1995.

Multi-scale MAP Estimation of High-Resolution Images

Shubin Zhao

Jiangsu Automation Research Institute,
42 East Hailian Road, Lianyungang, Jiangsu, China 222006
zhao_shubin@163.com

Abstract. In this paper, a multi-scale MAP algorithm for image super-resolution is proposed. It is well known that Reconstructing high-resolution(HR) images from multiple low-resolution(LR) images or a single one is an ill-posed problem. The main challenge is how to preserve edges in images while reducing noise. According to Bayesian approaches, which are popular and widely researched, solving this kind of problems is introducing prior knowledge about HR images as constraints and obtaining good HR images in some sense. In this paper, wavelet-domain prior distributions are concisely analyzed. And then, by introducing wavelet-domain Hidden Markov Tree-structured model(HMT) which accurately characterizes the statistics of most real-world images, reconstruction of HR images is reformulated as a multi-scale MAP estimation problem. For justification of this formulation, HMT is interpreted in the regularization framework, concisely and clearly. Experimental results are presented for assessment.

Keywords: Super-resolution, wavelet transform, MAP estimation.

1 Introduction

Images are produced in order to record or display useful information. Due to imperfections in the electronic or photographic medium, however, the recorded image often represents a degraded version of the original scene. The degradations may have many causes, but two types of degradations are often dominant: blurring and noise. Blurring is a form of bandwidth reduction of the image due to the imperfect image formation process. It can be caused by relative motion between the camera and the original scene, or by an optical system which is out of focus. In addition to these blurring effects, the recorded image is also corrupted by noises. These may be introduced by the transmission medium, the recording medium, measurement errors due to the limited accuracy of the recording system, and quantization of the data for digital storage. High-resolution(HR) images are desired in many applications such as remote sensing, printout from video, medical imaging, and so on. Using more advanced imaging devices can improve resolution of the images captured, but they are usually of high cost and sometimes even unavailable. So, for some applications it is important to obtain HR images from multiple low-resolution(LR) images or a single one, which is known as multi-frame super-resolution or single-frame super-resolution in literatures, respectively.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1059 – 1066, 2006.

Many algorithms have been proposed to address super-resolution problems. It is well known that super-resolution is an ill-posed problem. That is, according to the image formation process, there exist many HR images corresponding to observed LR images, so how to select the best HR image is a critical issue. A general idea is to regularize the problem by introducing the prior knowledge, and then obtain the MAP estimate of the HR image, which is the best approximation in some sense. In this kind of approaches, the main consideration is to preserve edges while reducing noise. Schultz and Stevenson[1-2] presented a Bayesian approach to obtain HR images from LR images, where the Huber-Markov Random Field(HMRF) is used as the prior distribution of the expected HR images. Hardie et al.[3] use a maximum aposteriori(MAP) framework for jointly estimating the registration parameters and the HR image for severely aliased observations, in which the Gibbs distribution is adopted as the prior knowledge. Ng et al.[4] develop a regularized constrained total least square solution to obtain a HR image. Nguyen et al.[5] have proposed circulant block preconditioners to accelerate the conjugate gradient descent method while solving the Tikhonov-regularized super-resolution problem. Other methods[6-10] are also used in solving related problems.

In our previous work[11], wavelet-domain HMT is used as the prior distribution to reconstruct HR images, but noise is not taken into consideration. In this paper, the method is extended by considering noise, and image super-resolution problem is formulated as multi-scale MAP estimation problem. For justification of the method, the HMT is interpreted in the regularization framework.

2 Problem Formulation

Generally, the relation between LR image and the corresponding HR image can be modeled as follows:

$$\boldsymbol{g}_k=\boldsymbol{H}_k\boldsymbol{f}+\boldsymbol{n}_k \quad 1\leq k\leq K \tag{1}$$

Where $\boldsymbol{g}_k$, $\boldsymbol{f}$ and $\boldsymbol{n}_k$ represent the observed LR image, the original HR image and noise, respectively; $\boldsymbol{H}_k$ models the blurring and down-sampling process, and hence the LR images are smaller than the HR one; K is the number of observed LR images. Note that images and noises are represented columnwise lexicographically ordered for convenience. For representational convenience, the above N equations can be grouped into one, then we get

$$\boldsymbol{g}=\boldsymbol{H}\boldsymbol{f}+\boldsymbol{n} \tag{2}$$

Mathematically, there is no significant difference between multi-frame super-resolution problem and single-frame one, and generally, they are all called by image super-resolution. Typically, $\boldsymbol{H}$ is ill-conditioned, and the existing noises make things even worse, so we cannot expect that there is an exact solution. In order to obtain the best approximation of the original HR image, the MAP method can be used. The MAP estimation of the unknown image $\boldsymbol{f}$ is done by maximizing the conditional probability density function of the original image given the observed LR image $P(\boldsymbol{f}|\boldsymbol{g})$. Based on Bayes rule, maximizing $P(\boldsymbol{f}|\boldsymbol{g})$ is equivalent to maximizing the function $P(\boldsymbol{g}|\boldsymbol{f})P(\boldsymbol{f})$. Here, $P(\boldsymbol{g}|\boldsymbol{f})$ is completely determined by the degradation process as formulated by (2), and $P(\boldsymbol{f})$ represents the prior knowledge about the ideal or expected HR images.

3 Multi-scale MAP Estimation

3.1 MAP Estimator in Wavelet Domain

In wavelet-domain, (2) is rewritten as

$$\boldsymbol{W}_g=\boldsymbol{H}_w\boldsymbol{W}_f+\boldsymbol{W}_n \tag{3}$$

Where $\boldsymbol{W}_g=\boldsymbol{A}_g\boldsymbol{g}$, $\boldsymbol{W}_f=\boldsymbol{A}_f\boldsymbol{f}$, $\boldsymbol{H}_w=\boldsymbol{A}_g\boldsymbol{H}\boldsymbol{A}_f^{-1}$, $\boldsymbol{W}_n=\boldsymbol{A}_g\boldsymbol{n}$. $\boldsymbol{A}_g$ and $\boldsymbol{A}_f$ are matrices for discrete wavelet transform(DWT) corresponding to $\boldsymbol{g}$ and $\boldsymbol{f}$, respectively.

For multi-scale MAP estimation, DWT of the searched HR image is given by

$$\operatorname{argmax}P(\boldsymbol{W}_f|\boldsymbol{W}_g) \tag{4}$$

Where $P(\boldsymbol{W}_f|\boldsymbol{W}_g)$ is the posterior probability. By Bayes rule, we have

$$P(\boldsymbol{W}_f|\boldsymbol{W}_g)= P(\boldsymbol{W}_g|\boldsymbol{W}_f)\ P(\boldsymbol{W}_f)/P(\boldsymbol{W}_g) \tag{5}$$

Note that $\boldsymbol{W}_f$ has no relation with $P(\boldsymbol{W}_g)$. So we get

$$\operatorname{argmax} P(\boldsymbol{W}_g|\boldsymbol{W}_f)\ P(\boldsymbol{W}_f) \tag{6}$$

Here, $P(\boldsymbol{W}_g|\boldsymbol{W}_f)$ is completely determined by the degradation process, and $P(\boldsymbol{W}_f)$ is the prior distribution of image. Suppose that $\boldsymbol{n}$ is i.i.d. Gaussian noise, and as a result of orthogonal DWT, $\boldsymbol{W}_n$ is also i.i.d. Gaussian noise. That is

$$P(\boldsymbol{W}_g|\boldsymbol{W}_f)=\frac{1}{\left(\sqrt{2\pi\sigma}\right)^N}e^{-\frac{\left\|H_W W_f-W_g\right\|_2^2}{2\sigma^2}} \tag{7}$$

Where σ^2 and N are variance and dimension of $\boldsymbol{W_n}$.

3.2 Modeling Natural Images in Wavelet Domain

In order to reduce noise, images are traditionally assumed to be smooth, but this assumption usually blurs the edges too much. So, effective image models should recognize the edge pixels, and hence can distinguish significant edges from smooth areas. Typically, the spatial structure of most natural images consists of smooth areas dispersed with sparse edges. Wavelet transform is a powerful tool to model the statistics of natural images. Because edges in an image are represented by few significant coefficients, the distribution of wavelet coefficients will be sharply peaked around zero, and has long tails corresponding to the edges, i.e. the density is non-Gaussian. Apart from this, the wavelet coefficients tend to propagate between scales, by which we mean that large/small values have persistence across scales. The wavelet-domain HMT model[12] models the non-Gaussianity as an independent mixture Gaussian. That is, random variable W_i is conditionally independent of all other variables given its state S_i. The marginal pdf of wavelet coefficients is given by

$$f_{W_i}(w_i) = \sum_{m=1}^{M} p_{S_i}(m) f_{W_i|S}(w_i \mid S_i = m) \tag{8}$$

Where $p_{S_i}(m) = p(S_i = m \mid \boldsymbol{W}, \boldsymbol{\theta})$, the probability $f_{W_i|S}(w_i \mid S_i = m)$ is Gaussian. Here, W_i and S_i are random variables for wavelet coefficient and its state, $\boldsymbol{W}$ wavelet coefficients, $\boldsymbol{\theta}$ HMT parameters.

The HMT captures the coefficient persistence across scales by linking the state variables across scales in a Markov tree, i.e. the state of a coefficient only depends on that of its parent node, as shown in Fig. 1.

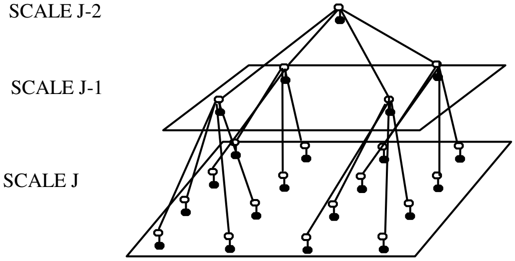


Fig. 1. An image quad-tree for Wavelet-Domain HMT model

If the three subbands are assumed to be statistically independent, the joint pdf of all wavelet coefficients can be computed as follows

$$f(\boldsymbol{W} \mid \boldsymbol{\theta}) = \prod_{1 \le i \le N} \sum_{m=1}^{M} p_{S_i}(m) f_{W_i|S_i}(w_i \mid S_i = m) \tag{9}$$

Now, by simple operations, (6) can be converted as

$$\underset{W_f}{argmin} \frac{1}{2\sigma^2} \left\| H_w W_f - W_g \right\|_2^2 - \sum_{i=1}^{N} \log\left(\sum_{m=1}^{M} p_{S_i}(m) f_{W_i|S_i}(w_i \mid S_i = m) \right) \tag{10}$$

4 Interpreting from the Viewpoint of Regularization

To show the underlying meaning, we further interpret and analyze the objective function in (10) from the viewpoint of regularization. In the regularization context, the first term reflects the fidelity of HR images to the observed LR images, and the second is the penalty function. To make things understood more easily, let the number of states of wavelet coefficients be two, namely, $M = 2$, which is commonly used to model most natural images. Suppose “1” means that the wavelet coefficient is small, and “2” means that the coefficient is large. We show $p_{S_i}(2)$ for the Lena image in Fig.2.

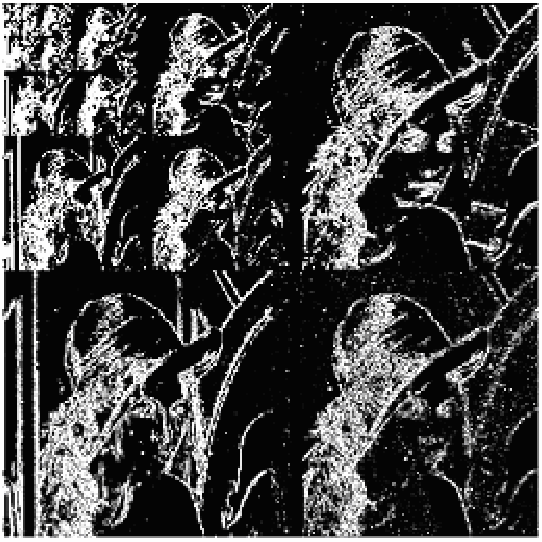


Fig. 2. The probability map: the intensity at each pixel equals to the probability that the corresponding wavelet coefficient is large

We can see from Fig.2 that the probability $p_{S_i}(2)$ has the capability of multi-scale edge detection. Let’s see a special case. If $p_{S_i}(n) = 1$, then

$$-\log\left(\sum_{m=1}^{2} p_{S_i}(m) f_{W_i|S_i}(w_i \mid S_i = m)\right) = \frac{w_i^2}{2\sigma_{i,n}^2} \tag{11}$$

Because $\sigma_{i,1}^2 << \sigma_{i,2}^2$, $p_{S_i}(1) = 1$ means that w_i is “suppressed”, and $p_{S_i}(2) = 1$ means that w_i is “encouraged”. So, this term works as a penalty function for the corresponding coefficient w_i, similar to the energy function in Gibbs distributions. Hence, noise will be effectively reduced in smooth areas, while significant information will be preserved around edges.

Following the idea shown above, we can simplify the optimization problem as follows

$$\underset{W_f}{\arg\min} \frac{1}{2\sigma^2}\left\| H_w W_f - W_g \right\|_2^2 + \sum_{i=1}^{N} \frac{w_i^2}{p_{S_i}(1)\sigma_{i,1}^2 + p_{S_i}(2)\sigma_{i,2}^2} \tag{12}$$

5 Algorithm Descriptions

Problem (12) can be rewritten as

$$\underset{W_f}{\arg\min} \frac{1}{2\sigma^2}(H_w W_f - W_g)^T (H_w W_f - W_g) + W_f^{\ T} Q^T Q W_f \tag{13}$$

Where $Q = diag(q_1, \cdots, q_i, \cdots, q_N)$ is a diagonal matrix, N is dimension of W_f. For scaling coefficients, $q_i = 0$; for wavelet coefficients,

$$q_i = 1 \Big/ \left(p_{S_i}(1)\sigma_{i,1}^2 + p_{S_i}(2)\sigma_{i,2}^2 \right) \tag{14}$$

From (13), it is easy to obtain

$$(H_W^T H_W + 2\sigma^2 Q^T Q) W_f = H_W^T W_g \tag{15}$$

In (15), $p_{S_i}(n)$ and W_f are coupled with each other. This problem can be solved by cyclic optimization similar to that in [11].

1) Set counter $k = 0$, initialize the posterior state probabilities $p_{S_i}^k(1) = 1$.

2) Solve problem (15) and obtain the solution W_f^k using steepest descent algorithm.

3) Compute $p_{S_i}^{k+1}(m) = p(S_i = m \,|\, W_f^k, \theta)$.

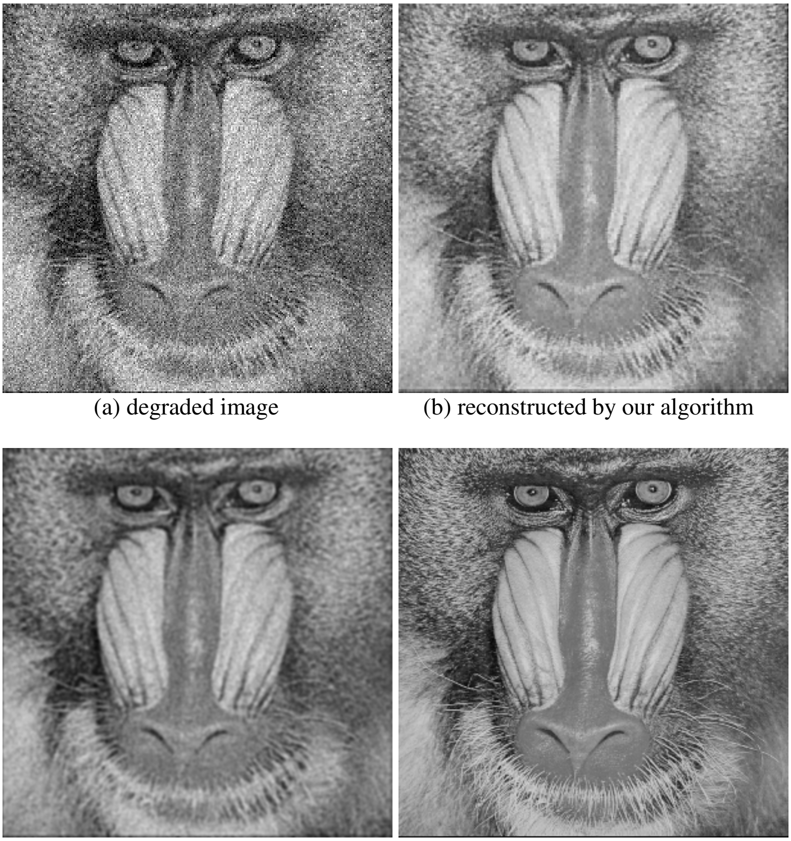
4) If $\frac{1}{N}\sum_{i=1}^{N} \left| p_{S_i}^{k+1}(1) - p_{S_i}^k(1) \right| < \varepsilon$, stop; else, set $k = k + 1$ and go to 2).

6 Experimental Results

Many experiments are made to demonstrate the effectiveness of the proposed algorithm. The peak signal-to-noise ratio (PSNR) is adopted to measure the image quality, and Schultz and Stevenson's method is used in comparison with our algorithm. In experiments, the Gaussian blurring kernel is used. Convolving the test HR images with the kernel, then subsampling and adding zero-mean Gaussian noise with standard deviation equal to 0.01, the resulting images are used as LR images. The Lena image is used to train HMT model. Four experimental results are shown in Table 1 for objective assessment and one experiment on the Baboon image is shown in Fig.3 for viewing.

Table 1. Experimental results for Lenna, Elaine, Baboon and Bridge

	Lenna	Elaine	Baboon	Bridge
Schultz'	22.5	23.6	20.1	22.9
Our algorithm	23.3	24.5	21.4	24.0



(a) degraded image

(b) reconstructed by our algorithm

(c) reconstructed by Schultz' method

(d) original image

Fig. 3. Experiment results for Baboon image: Schultz and Stevenson's method is used in comparison with our algorithm

7 Conclusions

This paper analyzes image statistics from the viewpoint of regularization, and then presents a multi-scale MAP estimation algorithm with wavelet-domain HMT model as the prior knowledge of the idea HR images. Experiments demonstrate that the reconstructed HR images have better quality.

References

1. Richard R. Schultz and Robert L. Stevenson, "A Bayesian Approach to Image Expansion for Improved Definition", IEEE Transactions On Image Processing, 1994, Vol. 3, No. 5, pp. 233-242.
2. R. R. Schultz and R. L. Stevenson, "Extraction of High-Resolution Frames from Video Sequences", IEEE Transactions On Image Processing, 1996, Vol. 5, No. 6, pp. 996-1011.

3. RC Hardie, KJ Barnard, and EE Armstrong, “Joint MAP Registration and High Resolution Image Estimation Using a Sequence of Undersampled Images”, IEEE Transactions On Image Processing, 1997, Vol. 6, No. 12, pp. 1621-1633.
4. M. K. Ng, J. Koo, and N. K. Bose, “Constrained total least squares computation for high-resolution image reconstruction with multisensors,” Int. J. Imaging Syst. Technol., vol. 12, pp. 35-42, 2002.
5. N. Nguyen, P. Milanfar, and G. Golub, “A computationally efficient super-resolution reconstruction algorithm,” IEEE Trans. Image Processing, vol. 10, No. 4, pp. 573-583, April 2001.
6. M. Elad and A. Feuer, “Restoration of a Single Super - Resolution Image from Several Blurred, Noisy and Under-Sampled Measured Images”, IEEE Transactions On Image Processing, 1997, Vol. 6, No.12, pp. 1646-1658.
7. S. Osher, L. I. Rudin, and E. Fatemi, “Nonlinear Total Variation Based Noise Removal Algorithms”, Physica D, 1992, Vol. 60, No. 3, pp. 259-268.
8. Murat Belge, Misha E. Kilmer, and Eric L. Miller, “Wavelet Domain Image Restoration with Adaptive Edge-Preserving Regularization”, IEEE Transaction On Image Processing, 2000, Vol. 9, No. 4, pp. 597-608.
9. S. Baker and T. Kanade, “Limits on Super-Resolution and How to Break Them”, IEEE Transactions On Pattern Analysis and Machine Intelligence, 2002, Vol. 24, No. 9, pp. 1167-1183.
10. S. Baker and T. Kanade, “Hallucinating Faces”, Proceedings of 4th IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, 2000, pp. 83-89.
11. Shubin Zhao, Hua Han and Silong Peng, “Wavelet-Domain HMT-Based Image Superresolution”, Proceedings of IEEE International Conference on Image Processing, 2003, pp. 953-956.
12. Matthew S. Crouse, Robert D. Nowak, and Richard G. Baraniuk, “Wavelet-Based Statistical Signal Processing Using Hidden Markov Models”, IEEE Transactions On Signal Processing, 1998, Vol. 46, No. 4, pp. 886-902.

Fast Transcoding Algorithm from MPEG2 to H.264

Donghyung Kim and Jechang Jeong

Department of Electrical and Computer Engineering, Hanyang University,
17 Haengdang, Seongdong, Seoul 133-791, Korea
{kimdh, jjeong}@ece.hanyang.ac.kr

Abstract. Video transcoding is the process of converting a video from one format into another. A format is defined by characteristics such as bitrate, framerate, spatial resolution and coding syntax. In this paper, we present an algorithm for transcoding from MPEG2 to H.264 in the spatial domain. For fast transcoding, we exploit three kinds of information included in an MPEG2 bitstream, which are coded macroblock type, coded block pattern and motion vector. According to the coded macroblock type and coded block pattern, we adaptively select the macroblock mode during the H.264 encoding process. Furthermore, the motion vector is also reused when an inter16x16 mode is selected as a macroblock mode. Simulation results show that the proposed transcoder dramatically reduces total transcoding time at comparable PSNR.

1 Introduction

Video transcoding alters the characteristics of a coded video [1]. The characteristics of the video stream include bitrate, framerate, spatial resolution, and coding standard. Transcoding can be classified into homogeneous and heterogeneous transcoding. The homogeneous transcoding method mainly attempts to reduce bitrate and divides into signal-to-noise ratio (SNR) [2], framerate [3] and spatial resolution [4] transcoding methods. The heterogeneous transcoding method attempts to change the coding standard of an input stream to another, such as MPEG2-to-MPEG4 transcoding [5].

As use of video content grows, transcoding becomes more important, especially to improve coding efficiency by transcoding from previous coding standards to H.264. The H.264 standard, however, shows substantial differences from previous video coding standards such as 4x4 integer transform, motion estimation using variable block sizes (VBS), multiple reference frames, etc. These new tools drastically increase the complexity of the H.264 encoder.

Several approaches have been proposed to reduce the complexity of a transcoder targeting the H.264 video standard. Bialkowski et al. proposed a fast transcoding algorithm between intra frames of the H.263 and the H.264 standard [6]. Their transcoding method exploits the fact that there exists a high correlation between the encoding processes of intra frames of two standards, and it has only about 4.2% of the complexity of intra mode selection in the H.264 encoding process. However, this method decreases visual quality up to 1.0 dB. Bialkowski et al. also proposed a fast transcoding algorithm between inter frames of the H.263 and the H.264 standard. By

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1067 – 1074, 2006.

reducing the number of search points for motion re-estimation, their transcoding method has only 10% of the complexity of a full search, and visual quality degrades by only about 0.1~0.5 dB [7].

Compared with H.263-to-H.264 transcoding, MPEG2-to-H.264 transcoding has more constraints because the MPEG2 standard is substantially different from the H.264 standard. Kalva, in his paper, introduced the issues of transcoding between the H.264 and the MPEG2 standards [8]. Chen et al. proposed an MPEG2-to-H.264 transcoding algorithm in the transform domain [9]. Their transcoder converts an 8x8 DCT in MPEG2 into a 4x4 integer transform in H.264, and reduces the spatial resolution simultaneously. As Chen's algorithm is carried out in the transform domain, the complexity of the transcoder can be reduced further. However it is not able to incorporate in-loop filtering and intra prediction, which should be applied in the spatial domain.

In this paper, we propose an MPEG2-to-H.264 transcoding method in the spatial domain. There are several reasons why the proposed transcoding is carried out in the spatial domain. First, we can not use in-loop filtering to eliminate blocking artifacts in the frequency domain because it is only applicable in the spatial domain. This problem leads to degradation of subjective quality. Second, the motion estimation using VBS in H.264 restricts the use of MC-DCT [10] which is used for transcoding in the frequency domain, especially when multiple reference frames are used. Third, the complexities of an 8x8 inverse DCT in MPEG2 and a 4x4 integer transform are much lower than those of motion estimation and macroblock mode selection in H.264.

An MPEG2-coded bitstream includes not only motion vectors but also the coded macroblock type (CMT) and the coded block pattern (CBP). The CMT and the CBP in MPEG2 are significantly related to the macroblock mode in H.264. Therefore we exploit the CMT and the CBP in order to adaptively select the macroblock mode during the H.264 encoding process. We also reuse the motion vector when an inter16x16 mode is selected as the macroblock mode in H.264.

2 Background

2.1 Comparison of MPEG2 and H.264

The MPEG2 coding standard has mainly targeted the broadcasting market with bitrates of around 4 Mbps for standard definition video. The H.264 coding standard is flexible and offers a number of tools to support a range of applications with very low as well as very high bitrate requirements. The H.264 standard gives perceptually equivalent video quality at 1/3 to 1/2 of the MPEG2 bitrate. There exist many differences between these two standards such as slice structure, transform kernel, in-loop filter, intra prediction, variable block size, and quarter-pixel-accuracy motion estimation/compensation. For more details, refer to [11].

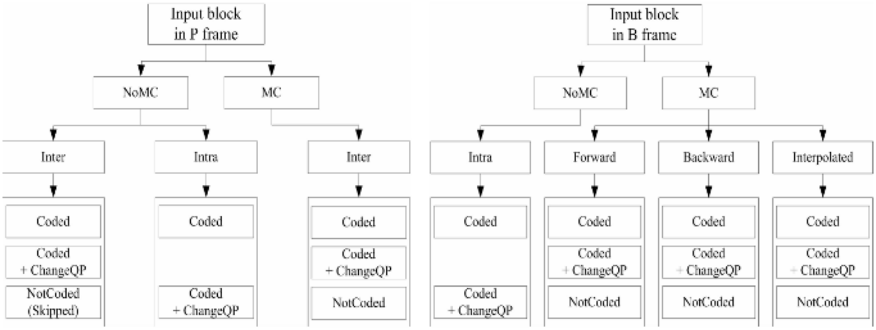
2.2 CMT and CBP in MPEG2

CMT refers to whether temporal prediction is used or not (Intra/Inter) during the MPEG2 encoding process, whether a motion vector other than zero vector exists or not (MC/No-MC), and which direction is used for temporal prediction (forward/backward/interpolated). The number of selectable CMTs depends on frame structures, i.e., I-, P- and B-frames. The CBP indicates which blocks are coded in a macroblock.

For a macroblock in I-frame, two CMTs can be chosen. They are Intra and Intra-ChangeQP. The former indicates that the macroblock is encoded as an intra block using the quantization parameter of the previous macroblock, and the latter indicates that the macroblock is encoded as an intra block using a new quantization parameter.

For a macroblock in P-frame, eight CMTs can be chosen as shown in Fig. 1(a). MC and NoMC indicate whether a motion vector other than the zero vector exists or not, respectively. Inter and Intra signify that temporal prediction is used and not used, respectively. NotCoded and Coded indicate whether all values after quantization are zeros or not, respectively.

For a macroblock in B-frame there are more selectable CMTs than in the P-frame because the B-frame uses more temporal prediction directions: forward, backward, and interpolated. Of all selectable CMTs, one skipped in the B-frame has a different meaning from one skipped in the P-frame. The block skipped in the P-frame is chosen when the macroblock has a zero motion vector and no block in the macroblock is coded, whereas the block skipped in the B-frame is chosen when the macroblock has the same motion vector and same temporal prediction direction as the previous one. Eleven selectable CMTs except the block skipped in the B-frame are described in Fig. 1(b).



(a) (b)

Fig. 1. Available CMTs in P- and B-frames (a) CMTs in P-frame (b) CMTs in B-frame expect the skipped block

2.3 Macroblock Mode in H.264

The H.264 standard allows the following macroblock modes: SKIP, inter16×16, inter16×8, inter8×16, inter8×8, intra16×16, intra8×8 and intra4×4. Furthermore, each block within inter8x8 can be divided into four sub-macroblock modes. The allowed sub-macroblock modes are: inter8×8, inter8×4, inter4×8 and inter4×4. Three intra modes have different prediction modes. There are four prediction modes in intra16x16, and nine prediction modes in intra8x8 and intra4x4.

2.4 Relationship Between the CMT/CBP and the Macroblock Mode

During the H.264 encoding process, macroblock mode selection is carried out by choosing the mode with the smallest rate-distortion (RD) cost. The RD cost function is as follows:

$$\text{RD cost} = \text{distortion} + \lambda\cdot\text{rate} \tag{1}$$

Among parameters of the RD cost function, the distortion is highly correlated with the CMT and the CBP in MPEG2. It is because they are also chosen depending on the distortion, that is to say, magnitude of the prediction error. For example, if every value of a macroblock during the transform and quantization process after prediction becomes zero, then Skipped is chosen as the CMT, and also if the distortion is large, then Intra may be chosen as the CMT. We can also estimate the prediction error of four sub-blocks within a macroblock using CBP. For example, if only one of all sub-blocks is not coded, then we may expect that only one sub-block has low prediction error. Therefore the CMT/CBP included in an MPEG2 bitstream is correlated to the macroblock mode in H.264, which can be effectively used for MPEG2-to-H.264 transcoding.

3 Proposed Algorithm

3.1 MPEG2-to-H.264 Transcoding

The proposed transcoder for MPEG2-to-H.264 transcoding consists of two parts: They are an MPEG2 decoding part and a H.264 encoding part. First of all, the proposed transcoder fully decodes an MPEG2-coded bitstream in the MPEG2 decoding part. Then, in an H.264 encoding part, the macroblock modes are selected adaptively by using CMT/CBP included in an MPEG2 bitstream, and motion vectors are also reused when an inter16x16 mode is chosen as a macroblock mode in the H.264 encoding process. Figure 2 illustrates the architecture of the proposed transcoder.

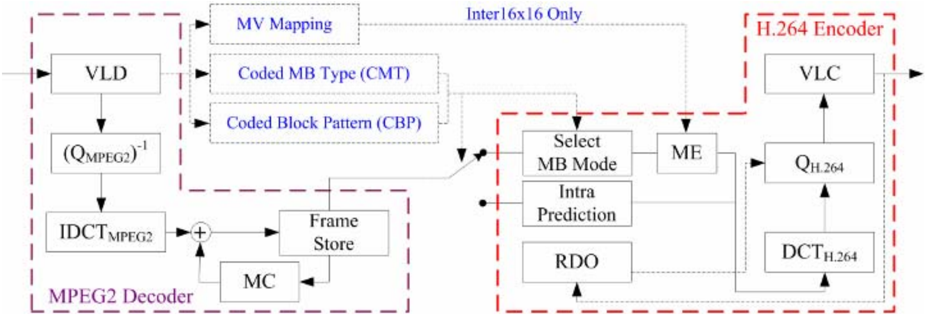


Fig. 2. The proposed transcoder which uses coded macroblock type, coded block pattern and motion vector included in an MPEG2 bitstream for MPEG2-to-H.264 transcoding

3.2 Adaptive Selection of the Macroblock Mode According to CMT and CBP

Although the proposed algorithm can be applied to both P- and B-frames, only the P-frame is considered in this paper in order to accomplish the transcoding targeting the H.264 baseline profile. Remember that the B-frame is not used in the H.264 baseline profile.

NoMC-Inter-NotCoded (Skipped) Macroblock. This CMT indicates that no block in the macroblock is coded, and it is usually chosen when the macroblock includes background or still objects. Since the distortion is very low in this case, we may expect that the RD cost of the corresponding macroblock is also very low in the H.264 encoding process. Therefore we only consider a SKIP mode as the macroblock mode in the H.264 encoding process.

NoMC-Intra-Coded Macroblock. This CMT indicates that the macroblock is intra-coded without motion estimation. This type is mainly chosen when the macroblock includes new objects which the reference frame does not have. Therefore, for the corresponding macroblock in the H.264 encoding process, we consider only two intra modes: intra4x4 and intra16x16 as the macroblock mode.

NoMC-Inter-Coded Macroblock. This CMT indicates that the macroblock has a zero motion vector. This type is mainly chosen when the macroblock includes objects with very fine motion. In this case, we may expect that the prediction error is very small even when motion estimation is carried out using a block size of 16x16. Therefore we only consider SKIP and inter16x16 as the macroblock modes in H.264.

MC-Inter-NotCoded Macroblock. This CMT indicates that no block within the macroblock is coded because the prediction errors become zeros after quantization. In other words, only a motion vector is coded in this CMT. It is mainly chosen when the motion vector of the macroblock is satisfactorily estimated from the reference frame using the block size 16x16. Therefore we also consider only SKIP and inter16x16 just like NoMC-Inter-Coded macroblock.

MC-Inter-Coded Macroblock. This CMT occupies most macroblocks. One motion vector and one or more blocks in the macroblock are coded in this macroblock type. When this type is chosen, we select the macroblock mode adaptively by using CBP in the H.264 encoding process. If two or more blocks in the macroblock are not coded, that is, the motion estimation is sufficiently correct for the block size of 16x8 or 8x16, we choose one of four macroblock modes: SKIP, 16x16, 16x8 and 8x16. If only one block is not coded, we may guess that the motion estimation is sufficiently correct for an 8x8 block. In this case we consider SKIP, 16x16, 16x8, 8x16 and 8x8 as the macroblock mode. Finally, if all blocks are coded, it means that the prediction error is large even for each 8x8 block. Therefore we consider all macroblock modes used for temporal prediction except intra modes: intra16x16 and intra4x4.

3.3 Reuse of Motion Vectors

The motion vector included in an MPEG2-coded bitstream is estimated only for a 16x16 block, and has half-pixel accuracy using a 2-tab filter. The motion vector in

H.264, however, is estimated for various block sizes, and has quarter-pixel accuracy using a 6-tap filter at the half-pixel position and a 2-tap filter at the quarter-pixel position.

The interpolated pixel value at the half pixel position is more important because it is reused for interpolating the pixel value at the quarter-pixel position. We, hence, had better re-estimate the motion vector at the sub-pixel position rather than reuse the one included in an MPEG2-coded bitstream.

Consequently, the proposed method only reuses integer-pixel-accuracy motion vectors when an inter16x16 macroblock mode is chosen in spite of the fact that the motion vectors included in an MPEG2 bitstream have half-pixel accuracy.

4 Simulation Results

Simulation has been performed on the first 100 frames of six standard video sequences in QCIF (176x144) format, which represents a different class of motion in each case. These include the Coastguard, Foreman, Stefan, Table Tennis, Hall monitor and Mobile sequences.

In our simulation, as shown in Table 1, the input MPEG2 bitstream is encoded at a bitrate of 500 Kbits/sec, and the output H.264 bitstreams are transcoded at various bitrates: 500, 400, 300, 200 and 100 Kbits/sec in order to compare the performance at different bitrates.

Table 1. Simulation conditions for MPEG2-to-H.264 transcoding

Comparison	MPEG2	H.264
CPU/Memory	Intel Pantium4 2.8 GHz / 512 MHz	
Codec	TM 1.1	JM 10.1
Profile/Level	Main/Main	Baseline/3.0
GOP/Structure	10/IPPP	10/IPPP
Bitrate(Kbits/sec)	500	500, 400, ···, 100

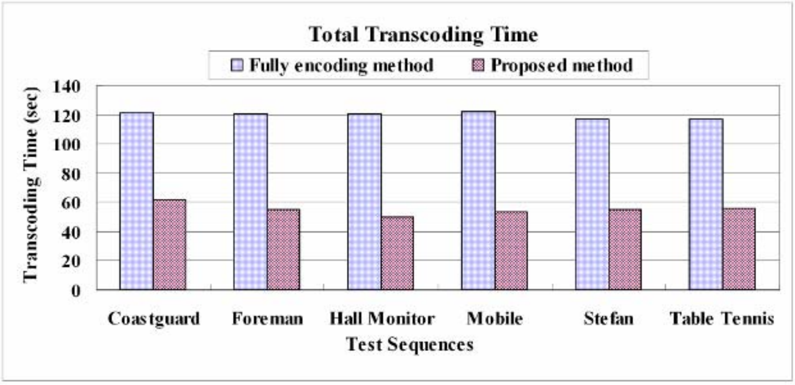


Fig. 3. Comparison of total transcoding time of the fully encoding method and the proposed method

Figure 3 compares the total transcoding time between the fully encoding method and the proposed methods. The fully encoding method indicates that the decoding process in H.264 is carried out without the use of side information such as motion vectors, CMT/CBP. As shown in these results, the proposed method saves about 60% of the total transcoding time compared with the fully encoding method, regardless of sequences. Table 2 compares the objective qualities and shows that the PSNR drop is only 0.03 dB on average when using our transcoder.

Table 2. PSNR comparison between the fully encoding and the proposed method

Sequences	Comparison	Bitrate Ratios (MPEG2/H.264)				
		500/100	500/200	500/300	500/400	500/500
Coastguard	Fully Enc.(1)	29.12	30.72	31.49	31.97	32.26
	Proposed (2)	29.12	30.70	31.48	31.95	32.25
	(2)-(1)	0.00	0.02	0.01	0.02	0.01
Foreman	Fully Enc.(1)	31.99	34.34	35.11	35.44	35.63
	Proposed (2)	31.97	34.29	35.06	35.40	35.60
	(2)-(1)	0.02	0.05	0.05	0.04	0.03
Hall Monitor	Fully Enc.(1)	34.18	36.10	36.68	36.97	37.10
	Proposed (2)	34.09	36.06	36.66	36.95	37.08
	(2)-(1)	0.09	0.04	0.02	0.02	0.02
Mobile	Fully Enc.(1)	22.52	24.28	25.10	25.47	27.10
	Proposed (2)	22.50	24.28	25.10	25.45	27.08
	(2)-(1)	0.02	0.00	0.00	0.02	0.02
Stefan	Fully Enc.(1)	23.92	26.83	27.93	28.42	28.72
	Proposed (2)	23.87	26.80	27.87	28.38	28.70
	(2)-(1)	0.05	0.03	0.06	0.04	0.02
Table Tennis	Fully Enc.(1)	30.82	32.51	33.19	33.56	33.78
	Proposed (2)	30.77	32.45	33.16	33.52	33.74
	(2)-(1)	0.05	0.06	0.03	0.04	0.04

5 Conclusions

The proposed algorithm is for heterogeneous transcoding from MPEG2 to H.264 in the spatial domain. For fast transcoding, we exploit three kinds of information included in an MPEG2-coded bitstream, which are motion vector, coded block pattern and coded macroblock type. We adaptively select the macroblock mode using the coded block pattern and coded macroblock type during the H.264 encoding process, and also reuse the motion vector when an inter16x16 mode is chosen as a macroblock mode.

Simulation results show that the proposed method can save about 60% of the total transcoding time regardless of sequences, yet the average PSNR drop is only 0.03 dB on average. This reduction of the transcoder complexity helps in real-time implementation of MPEG2-to-H.264 transcoding.

Acknowledgement

This research was supported by Korean Film Council under Advanced Digital Imaging Technology Development Project.

References

1. Vetro, A., Christopoulos, C., Sun, H.: Video Transcoding Architectures and Techniques - An Overviews. IEEE Signal Processing Magazine (2003) 18-29
2. Nakajima, Y., Hori, H., Kanoh, T.: Rate Conversion of MPEG Coded Video by Re-quantization Process. ICIP'95, Vol. 3. (1995) 408-411
3. Chen, M.J., Chu, M.C., Pan, C.W.: Efficient Motion Estimation Algorithm for Reduced Frame-Rate Video Transcoder. IEEE Trans. Circuits and Syst. for Video Technol., Vol. 12. (2002) 269-275
4. Lee, Y.R., Lin, C.W., Chen, Y.W.: Computation Reduction in Cascaded DCT Domain Video Downscaling Transcoding. ISCAS'03, (2003) 860-863
5. Xin, J., Sun, M.T., Chun, K.: Motion Re-estimation for MPEG2 to MPEG4 Simple Profile Transcoding. PV'02, Pittsburgh, (2002)
6. Bialkowski, J., Kaup, A., Illgner, K.: Fast Transcoding of Intra Frames Between H.263 and H.264. ICIP'04, Vol. 4. (2004) 2785-2788
7. Bialkowski, J., Menden, M., Barkowsky, M., Illgner, K., Kaup, A.: A Fast H.263 to H.264 Inter-Frame Transcoder with Motion Vector Refinement. PCS'04, (2004)
8. Kalva, H.: Issues in H.264/MPEG2 Video Transcoding. CCNC'04, (2004) 657-659
9. Chen, C., Wu, P.H., Chen, H.: MPEG2 to H.264 Transcoding. PCS'04, (2004) 15-17
10. Chang, S.F., Messerschmitt, D.G.: Manipulation and compositing of MC-DCT compressed video. IEEE Journal on Selected Areas in Communications. Vol. 13. (1995) 1-11
11. ISO/IEC 14496-10: Version 3 of H.264/ AVC (2004)

A New Display-Capture Based Mobile Watermarking System

Jong-Wook Bae, Bo-Ho Cho, and Sung-Hwan Jung

Dept. of Computer Engineering, Changwon National University
9 Sarim-dong, Changwon, Gyeongnam, 641-773, South Korea
bae@changwon.ac.kr, {chobh, sjung}@sarim.changwon.ac.kr

Abstract. In this paper, we propose a new display-capture based mobile watermarking scheme that basically uses a generated image for watermark embedding. The proposed scheme is a new mobile watermarking concept. Instead of using a given original image in the conventional watermarking, it uses a proper generated image for inserting information. Some image distortions may occur in the display-capture environment of a mobile phone. They make a problem in the watermark decoding process. To solve the problem, we use hardware and software approaches and also model the image distortion that usually may happen in the mobile phone. As experimental result, we could embed typical MMB (Mobile Merchandise Bond) information into the background image and generate a watermark-embedded image. With preprocessing to reduce distortions, we could extract the inserted information from the watermark-embedded image captured by a PC camera. Therefore, we could confirm availability of a new mobile watermarking in the display-capture environment.

Keywords: Mobile watermarking, Display-capture, Image distortion, MMB (Mobile Merchandise Bond) service, MMB image generation.

1 Introduction

Various mobile services have been developed for the mobile environment. There is MMB (Mobile Merchandise Bond) service among the mobile services [1]. Let us look over recently commercialized some MMB services [2]: 1) the usage of bar code - the case that the bar code of merchandise bond would be taken by a mobile device such as a mobile phone, and a customer has to exchange the bar code image for an actual merchandise bond in a member store; 2) the usage of special card - the case that the customer must hold a merchandise bond card as a specially distinguished card; 3) the usage of special device - the case that the usage of merchandise bond is only possible by the special IC chip built-in a mobile device. These methods have the convenience that a customer can download a merchandise bond freely through wire or wireless line. However, when the customer uses the merchandise bond, he or she must exchange it into the actual merchandise bond or must have a special card. Such things are annoying for customers. To solve such current problems of MMB service, we propose a new display-capture based mobile watermarking system.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1075 – 1084, 2006.

Most of existing watermarking technologies are related with the research of robust watermarking for copyright protection [3, 4]. Fragile watermarking has been studied as an authentication method whether the contents are modified or not [5]. Recently print-scan watermarking has also been studied. It can prevent forgery for printed contents [6, 7]. However, the research about the mobile authentication technology in the mobile environment is just in the initial stage, when it is compared with the current fast diffusion of mobile devices. Furthermore, most of the mobile authentication is done by not the watermark-based technology but the bar code-based technology and others.

In this paper, we propose a new mobile watermarking scheme based on the display-capture. Most of existing watermarking schemes use an original image for embedding a watermark, and focus on imperceptibility of the watermark. But the proposed mobile watermarking scheme is a new concept, which uses a watermark-embedded image for MMB in the mobile environment. In our system, a customer can straightforwardly use a watermark embedded MMB displayed in the mobile device in the member store without any others things. We also study about the display-captured distortions that may happen at the display-capture process in the mobile watermarking system.

2 Display-Capture Based Mobile Watermarking System

The display–captured based mobile watermarking system is described in the Fig. 1. A MMB issuing company can generate the MMB image embedded MMB information as a watermark. And the company can simply register it to DRM system. A customer can buy a MMB and download it through the Internet. The customer displays the MMB image in his or her mobile device, and then the MMB image of the mobile device is captured by a PC camera in a member store. The MMB information is extracted from the captured image, and the authentication through DRM system is occurred. Finally the transaction of MMB can be done.

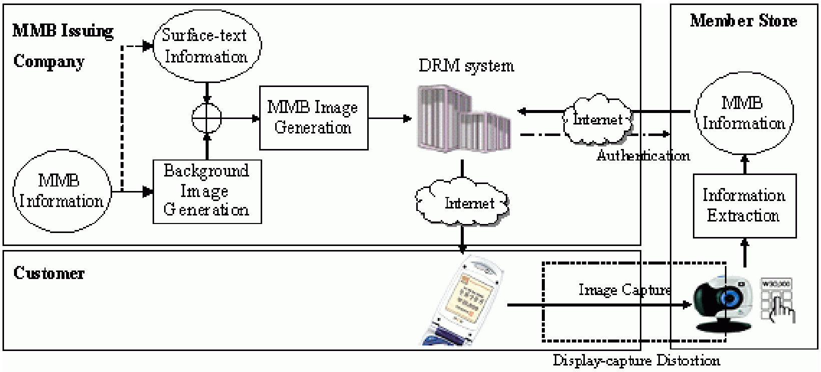


Fig. 1. General block diagram of mobile watermarking system

In this paper, we focus on the MMB image generation, display-capture distortion, and information extraction from the captured MMB image. Now we assume that the DRM system has a simple function. The detailed functions of DRM system can be handled in the next further research.

2.1 Background Image Generation

A Mobile Merchandise Bond (MMB) image consists of a background image and surface-text information. The background image is a generated image embedded a watermark by using the unit-pattern. The surface-text is visual text information on the MMB image. The process of the background image generation is shown in Fig. 2.

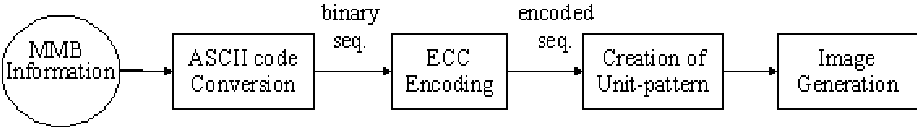


Fig. 2. Generation process of background image

From MMB information such as the company name, the date of issue, the serial number, and the amount of money, etc., we get a binary sequence by ASCII code conversion. The next step is to apply error-correction code (EEC) to the sequence. The unit-pattern can be created according to the encoded sequence. Finally the background image can be generated based on unit-patterns.

An example of the generated background image is shown in Fig. 3. A unit-element consists of 2x2 pixels. It can be used to express one intensity level among k-step's intensity levels. For the easy recognition of the unit-element, the background image has the separate lines of 2x2 pixels size among unit-elements. And a unit-pattern consists of 4 unit-elements.

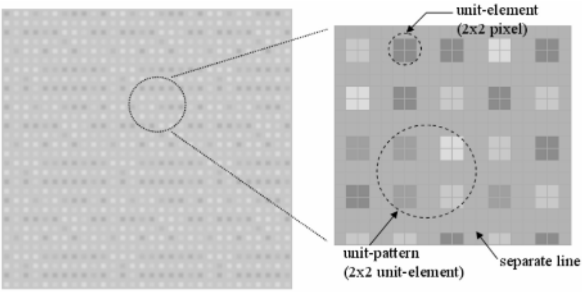


Fig. 3. Example of generated background image

In addition, we need a technique to prevent image distortion caused by the image capture of a PC camera. We also apply ECC to generate a robust sequence against the occurrence of error during extraction of the sequence. We also insert the sequence iteratively in the whole image and finally generate the background image. We use

Viterbi algorithm as ECC generation in this research. Viterbi EEC encoder makes the original sequence double in the length when the constraint length is 3 and the code rate 1/2 [8]. Therefore, we can calculate the insertion capacity by equation (1).

$$\text{Insertion Capacity} = \frac{1}{2}\left\lfloor \frac{1}{8}\frac{M}{m} \right\rfloor \cdot \left\lfloor \frac{1}{8}\frac{N}{n} \right\rfloor \cdot \log_2 k^4 \;\; [bit] \qquad (1)$$

Where M, N are the width and height of an image. m, n indicate iteration numbers of inserting MMB information in the width and height direction respectively. k presents the k-step's intensity level that the unit-element can express.

2.2 MMB Image Generation

A MMB image consists of a background image for information embedding and a text for displaying information visually on the background image. Let us call the text as the surface-text information. If we overlap the surface-text information directly on the background image, some parts of the background image are lost, so that the decoding process can be difficult. Therefore, we can use a color image to express the surface-text information without the loss of MMB information inserted in the background image. The process of MMB image generation is depicted in Fig. 4.

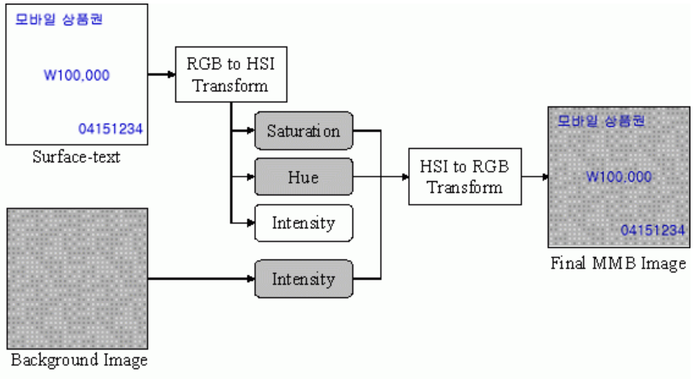


Fig. 4. Process of MMB image generation

We create the color image which displays the surface-text information and transform its RGB color space to HSI color space. We replace its intensity channel with I channel of the background image and transform HSI to RGB. Finally the MMB image is generated like the most right figure of Fig. 4.

3 Image Distortion of Display-Capture

Fig. 5 shows the general processing steps of the MMB in the display-capture environment. First, a MMB image is generated based on the MMB information. Then, the generated MMB image is displayed on the LCD (Liquid Crystal Display) of a mobile device such as a mobile phone. The displayed MMB image is captured along with the distortion of the display-capture process [9-12]. Through the watermark extraction process, the MMB information is extracted from the captured MMB image.

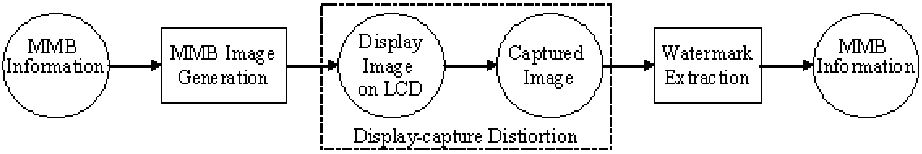


Fig. 5. General block diagram of mobile merchandise bond process

If the extracted MMB information is certified through the DRM system, the actual commercial transaction may happen. In the display-capture process, the causes of image distortion are as follows: surrounding lighting conditions, the surface reflection of a mobile device, the geometric distortion by the natural habit of a customer, and the quality of camera lens, etc. We deal with hardware and software approaches to solve the image distortions that may happen by these causes in the display-capture environment.

With hardware approach, we can reduce the effect of external unnecessary lighting by installing reflection covering (RC) or blackout boxing (BB). We can reduce the distortion of external unnecessary lighting with them. However, when we capture the MMB image on a mobile device, a distorted image with un-uniform lighting is captured because the backlight of the mobile phone sheds light from top to bottom direction. With software approach, we can restore the distortion of backlighting by compensating it based on modeling the distortion of backlighting with the quadratic equation, $y=ax^2+bx+c$ [13, 14].

4 Information Extraction from MMB Image

The decoding process of a MMB image, watermark-embedded image, is depicted in Fig. 6. A PC camera may capture the MMB image displayed on the LCD of a mobile phone. The preprocessing can compensate the distortions caused in the display-capture process mentioned chapter 3.

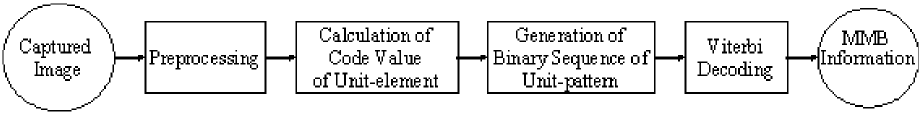

Captured Image
Preprocessing
Calculation of Code Value of Unit-element
Generation of Binary Sequence of Unit-pattern
Viterbi Decoding
MMB Information

Fig. 6. Decoding process of watermark embedded image

To get MMB information from the captured MMB image, real decoding steps are needed, including calculation of code value of unit-element, restoration of the binary sequence of unit-patterns, and Viterbi decoding.

4.1 Preprocessing

A captured MMB image may have various distortions. To solve the distortions, we apply image recovering technique mentioned in chapter 3. We here only show the backlighting distortion as an example. A MMB image captured from the LCD of a mobile phone may have the distortion like Fig.7. Intensity values of upper area are different from that of lower area. Upper area of the captured image is relatively brighter than lower one. So the intensity value of all pixels at a unit-element throughout the whole image may be different.

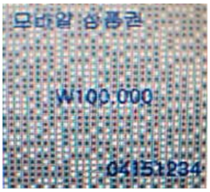


Fig. 7. Captured image with distortions

To solve this problem, we can apply the distortion model of the mobile phone's backlighting which was discussed in section 3. In case of geometric distortion like rotation, we can also restore the MMB image by the inverse transform.

4.2 Calculation of Code Value

We transform each unit-element into binary code as Fig. 8. To cope with the distortion still remained in the MMB image after the above mentioned preprocessing step, we additionally use the higher resolution capture and median filtering technique as follows: We captures the watermark embedded image displayed on LCD as 2 times high resolution. So the 2x2 pixels sized unit-element of the captured image has 4x4 pixels. We sort all 16 pixels in a unit-element by intensity value and select the median pixel value V_u. Also we sort all 128 pixels in a separate line enclosing the unit-element by intensity value and select the median pixel value V_s. Then we decide code value like equation (2).

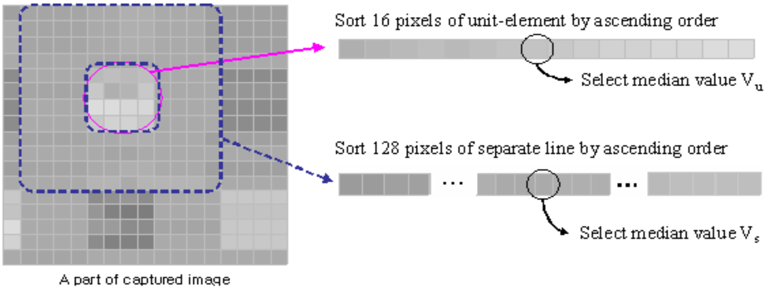

Sort 16 pixels of unit-element by ascending order
Select median value V_u
Sort 128 pixels of separate line by ascending order
···
···
Select median value V_s
A part of captured image

Fig. 8. Code value selection of unit-element

$$code\,value\,of\,unit-element\begin{cases} V_u > V_s, & 1 \\ otherwise, & 0 \end{cases} \tag{2}$$

According to mapping the pixel value of a unit-element into code value, the unit-patterns of the watermark-embedded image can be changed to a binary sequence. Then, finally we can recover the MMB information of a MMB image through the Viterbi decoder.

5 Experimental Results and Discussions

In this paper, we used a mobile phone of model number LG-SD9230 to display a watermark-embedded image, MMB image. And this mobile phone support 160x120 DOT resolution. We used VIJE Talk CCD PC camera to capture an image displayed on LCD. We decided the size of the MMB image as 120x120, considering general LCD's size of mobile phones. And MMB information consists of the company name of issue, the date of issue, the serial number, and the amount of money, etc.

We generated watermark-embedded image with ECC and without ECC, respectively and tested them. The results are shown in Fig. 9, 10 in case that an intensity level, k is 2-steps.

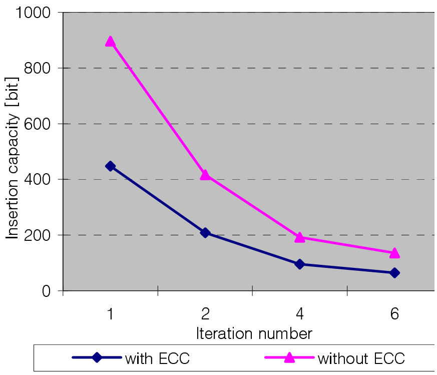


Fig. 9. Insertion capacity with ECC and without ECC

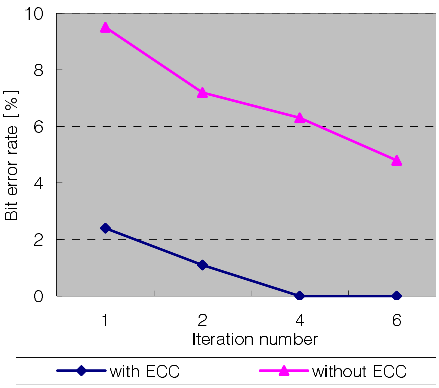


Fig. 10. Bit error rate with ECC and without EEC

In Fig 9, as the iteration number is increasing, the insertion capacity is decreasing. In Fig 10, as the iteration number is increasing, the Bit Error Rate is decreasing. When the iteration number reaches to 4, the Bit Error Rate becomes zero. It means that the complete recover of information is possible.

The following images are related with the hardware approach for the image distortion of display-capture. Fig. 11 is the distorted image captured in the open environment. Fig. 12, 13 are the captured images with the reflection cover and the blackout box environment, respectively.

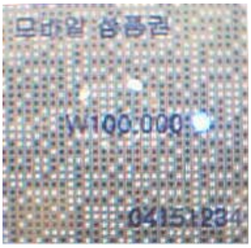


Fig. 11. Captured image with open environment

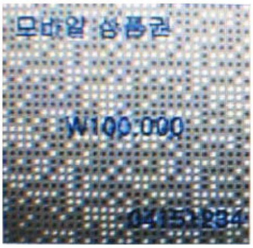


Fig. 12. Captured image with reflection cover

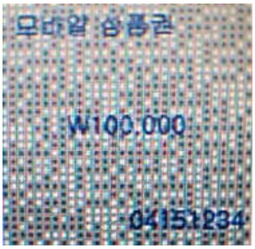


Fig. 13. Captured image with blackout box

Fig. 14 is the plot of comparison of the measured intensity value, $\hat{y}_i$ and model value. We experimentally get the coefficient of the quadratic equation, $y=ax^2+bx+c$: a = 0.0041, b = -0.2381, c = -70.2435, respectively.

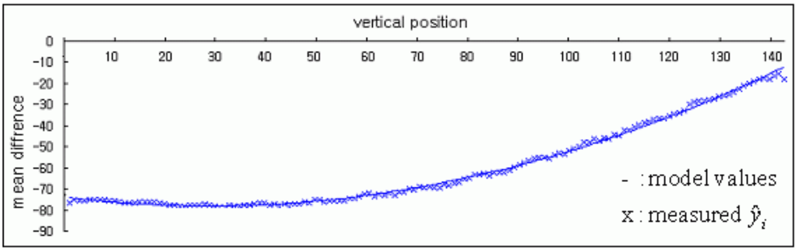


Fig. 14. Comparison of model value and measured $\hat{y}_i$

Table 1 shows the experimental results and information for various distortions. In the open environment, it was difficult to extract the text information from the captured image for a lot of external noises.

Table 1. The experimental results and information for various distortions (RC: Reflection Covering, BB: Blackout Boxing)

Approach	Distortions	Condition	Light	Extraction
Hardware approach	Reflection	Open environment	ON, OFF	No
		RC	ON	Yes
	Angle of view	RC	ON	Yes
		BB	ON	Yes
Software approach	Brightness	No correction, RC, BB	OFF	No
		No correction, RC, BB	ON	Yes
		Correction, RC	OFF	No
		Correction, BB	OFF	Yes
	Rotation	No correction, BB	ON	No
		Correction, BB	ON	Yes
	Brightness + Rotation	No correction, BB	ON	No
		Correction, BB	ON	Yes

In the reflection cover and blackout box environment, it was possible to extract the MMB information from the captured image with phone light on (40 lx). However, we could not extract the MMB information from the captured image with light off (2~3 lx) because the captured image is dark. By correcting the captured image with light off, we could not extract the MMB information under the reflection cover, but could extract the MMB information from the captured image under of the blackout box environment. This fact shows that the reflection cover environment is more sensitive as to the light than the blackout box because the angle of view of the reflection cover is the wider than that of blackout box.

Fig. 15 shows a real example of MMB image that has the geometric distortion and also the distortion of back lighting of a mobile phone. In this case, first, we restored the geometric distortion using the inverse function of geometric distortion and then, for the back lighting distortion, we corrected the pixel values of the distorted image based on the distortion model like Fig. 14 [14]. Fig. 16 shows the result of extracted MMB information from the captured image in Fig. 15 through the process of Fig. 6.

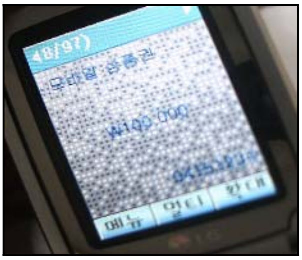
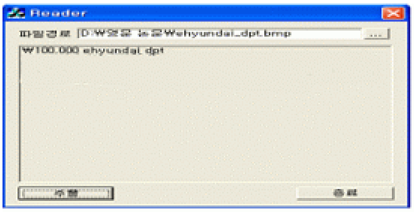


Fig. 15. Example of MMB image



파일경로 D:₩[illegible]₩ehyundai_dpt.bmp

₩100,000 ehyundai_dpt

Fig. 16. Example of Extracted MMB information

6 Conclusions

In this paper, we present a new research for the display-capture based mobile watermarking. This research deals with the generation technique of a watermark-embedded image for applying in the display-capture environments. The proposed watermark-embedded image generation is different from most of existing watermarking schemes. The proposed technique is a new concept that uses the generated image instead of the given image. It generates a proper MMB image for the watermark to be inserted.

As experimental results, we could embed typical MMB information and generate a watermark-embedded image. And we could extract information from the watermark-embedded image captured by the PC camera. So we could confirm availability of the generation of watermark-embedded image in the display-capture environment.

Also, we studied the distortions for display capture-based mobile watermarking. First, we solved the distortions that can happen with the light influence of outside by hardware approach. Then, we applied the software approach to solve distortion of the backlighting of the mobile phone and geometric distortion. As the result of our research, we could know that these two approaches could settle effectively the distortions in the display-capture environment. In the open environment, we could not extract the MMB information because of a lot of noises. Under the reflection cover and blackout box environment, we could extract the text information with phone light on.

In further research, we will study the image generation with high complex background and the extraction of MMB information from the captured image and handle various distortions that may happen in the display-capture environment.

Acknowledgments

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2005-041-D00757).

References

1. http://k-merce.magicn.com/index.asp?code=IB00000 (2006)
2. http://gift.ez-i.co.kr/NASApp/web/info/info_use_online.jsp (2006)
3. Ching-Yung Lin, Min Wu, Jeffrey A. Bloom, Ingemar J. Cox, Matt L. Miller, Yui Man Lui: Rotation, Scale, and Translation Resilient Public Watermarking for Image. Proceedings of SPIE, Vol. 3971, No. 2 (2000) 90-98
4. M. Kutter: Watermarking Resisting to Translation, Rotation and Scaling. Proceedings of SPIE: Multimedia systems and applications, Vol. 3528, Boston (1998) 423-431
5. E. T. Lin and E. J. Delp: A Review of Fragile Image Watermarks. Proceedings of the Multimedia and Security Workshop (ACM Multimedia '99) Multimedia Contents (1999) 25-29
6. C. Y. Lin, and S.F. Chang: Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process. ISMIP 99 (1999)
7. Jong-Wook Bae, Sin-Joo Lee, Sung-Hwan Jung: LPM-Based Digital Watermarking for Forgery Protection in Printed Materisls. Journal of Korea Multimedia Society, Vol.8, No.11 (2005)
8. Viterbi Algorithm, http://lcmwww.epfl.ch/APPLETS/VITERBI /Viterbi.html (2006)
9. J. Stach, T. Brundage, B. Hannigan, B. Bradley, T. Kirk, H. Brunk: On the Use of Web Cameras for Watermark Detection. Security and Watermarking of Multimedia Contents IV, Vol. 4675 (2002) 611-620
10. B. Perry, B. MacIntosh, D. Cushman: Digimarc MediaBridge - The Birth of a Consumer Product, from Concept to Commercial Application. SPIE Electronic Imaging 2002, Security and Watermarking of Multimedia Content IV, San Jose, CA (2002)
11. G. T. Han, W. Y. Kim: A Calibration Coefficient Auto Extraction Method for Compound Distortion Image. KICS Journal, Vol. 26 No.3B, (2001) 302-314
12. W. P. Yu, Y. K. Chung: An Image Warping Method for Implementation of an Embedded Lens Distortion Correction Algorithm. KIPS Journal, Vol. 10B, No. 04 (2003) 373-380
13. Steven C. Chapra, Raymond P. Canale: Numerical Methods for Engineers, 2nd edition, McGraw-Hill, Inc. (1988)
14. Bo-Ho Cho, Jong-Wook Bae, Jae-Young Lee, Jeng-Hae Youn, and Sung-Hwan Jung: A Study on Image Distortion for Display-capture based Mobile Watermarking. Asian Language Processing and Information Technology, Tashkent (2006)

Adaptive Data Hiding for Images Based on Harr Discrete Wavelet Transform

Bo-Luen Lai and Long-Wen Chang

Department of Computer Science
National Tsing Hua University
HsinChu, Taiwan

Abstract. In this paper, we propose an adaptive data hiding method in the frequency domain. We divide the original image into 8*8 sub-blocks and use Haar Discrete Wavelet Transform (HDWT) to decompose each block to obtain LL1, HL1, LH1 and HH1 bands. Since Human eyes are insensitive to the edge region we embed more data when the band LL1 is complex. A data hiding capacity function is used to analyze the complexity of LH1, HL1 and HH1 bands. If these three subbands are complex, the LL1 band is decomposed and more data bits are embedded in the further decomposed LH2, HL2 and HH2 bands. From the experiments, we find that the proposed method outperforms other methods in both data hiding capacity and image quality.

1 Introduction

Data hiding [1, 2] is a technology to hide secret information for multimedia personalization, interaction, protection and fighting piracy. The file that is used to hide secret information is called a “cover-file”. The cover-file can be all kind of files, like videos, images, or audios, to embed our secret information. If we choose an image to hide our information, we call the original image as a “cover-image” and the “stego-image” after embedding the secret information. The best advantage of using data hiding is that it could hide the important information into a common multi-media file to avoid any attack. Many data hiding techniques are proposed before. The common method is replacing n least significant bits (n-LSB) [3] of the pixel of the cover image directly. The method is simple, quick, and efficient. When n is small, the stego-image seems the same as the cover-image. In our experiments, if we replace 3-LSB of the cover-image, we can get good quality of stego-image. However, the LSB method has drawbacks. The LSB method is a non-adaptive method of data hiding for fixed capacity whether the cover image is smooth or complex. It is known that all pixels of the cover image cannot embed the same length of data. If we change too many LSBs on the pixels of the smooth region, we can detect the difference between the cover image and the stego-image. Wu and Tsai et al. [4] don’t change LSB of the pixel directly. They change the LSB of the difference value of consecutive pixels. They divide the cover image into 2*1 non-overlapping blocks. Each block is classified by its difference value. A small difference value means the block in smooth region, and a big one means in edge block. The embedded amount of data depends on the characteristics of each block.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1085 – 1093, 2006.

In this paper, we propose an adaptive data hiding method in the frequency domain. We divide the original cover image into 8*8 sub-blocks and use Haar Discrete Wavelet Transform (HDWT) to decompose each block to obtain sub-bands, LL1, HL1, LH1 and HH1. A data hiding capacity function is proposed to analyze HL, LH and HH bands to decide whether the LL1 band is suitable for more data hiding or not. If it is complex it is further decomposed and more data are embedded in LH2, HL and HH bands.

2 The Proposed Method

1D Haar Discrete Wavelet Transform [5] (1D-HDWT) is one of the most common DWT. Its basic idea uses a *average value* **S** and successive levels of *details*, $\mathbf{D_1}$, $\mathbf{D_2}$, $\mathbf{D_3}$... to express a discrete signal. 2D-HDWT can be done by using a sequence of column 1D-HDWTs and row 1D-HDWTs. Suppose the original signal is $S_i^0 = \{S_0^0, S_1^0, ..., S_{k-1}^0\}$, we can get low frequency signal $S_i^1 = \{S_0^1, S_1^1, ..., S_{\frac{k}{2}-1}^1\}$ and high frequency signal $D_i^1 = \{D_0^1, D_1^1, ..., D_{\frac{k}{2}-1}^1\}$ by the following equation:

$$\begin{cases} D_i^1 = S_{2i}^0 - S_{2i+1}^0 \\ S_i^1 = \left\lfloor \dfrac{S_{2i}^0 + S_{2i+1}^0}{2} \right\rfloor \end{cases} \tag{1}$$

We can do inverse HDWT. By the following equation:

$$\begin{cases} S_{2i}^0 = S_i^1 + \left\lfloor \dfrac{D_i^1 + 1}{2} \right\rfloor \\ S_{2i+1}^0 = S_{2i}^0 - D_i^1 \end{cases} \tag{2}$$

where i = 0, 1, 2…, k/2-1, k is the length of input samples and $\lfloor . \rfloor$ denotes the flooring function. The advantage of HDWT is that it is very simple and the forward transform and its inverse is integer-to-integer process, which is often used for lossless image compression [6]. The process of 2DHDWT can be illustrated in Figure 1. There is a problem when we embed the data in the frequency domain. Some coefficients will make underflow/overflow after embedding the data on these coefficients (at 8-bit gray level image, underflow means the pixel value smaller than 0 and overflow means the pixel value greater than 255.). For example, suppose we take 2D-HDWT of $\begin{bmatrix} 25 & 2 \\ 18 & 6 \end{bmatrix}$, we can get $\begin{bmatrix} 12 & 17 \\ 1 & 11 \end{bmatrix}$. If we modify 17 to become 22, we have $\begin{bmatrix} 12 & 22 \\ 1 & 11 \end{bmatrix}$. After inverse 2D-HDWT of the matrix, we have $\begin{bmatrix} 27 & -1 \\ 21 & 4 \end{bmatrix}$. There is a value below 0. That is, an underflow occurs. This status usually happens when the original pixel value is close to 0 or 255. Therefore, we adjust the histogram of the cover image to solve this problem before doing HDWT. We can modify the pixel values that are close to 0 or 255 for avoiding overflow or underflow. The method is as following:

If F (i,j) ≧ 255-G/2, modify F(i,j)= F(i,j) – G/2;
If F (i,j) ≦ G/2, modify F(i,j) = F(i,j) + G/2,

where F(i,j) is the pixel value at the location of (i,j) of an image F. G is the argument to modify histogram of an image. We usually set G to 30 to avoid underflow or overflow. After the histogram modification, we can decompose the image with two-dimensional HDWT to be LL1, LH1, HL1 and HH 1bands. Each coefficient of HL1, LH1, and HH1 bands means vertical, horizontal, and diagonal changes. If the absolute value of the wavelet coefficient in these bands is large, it is possible to be an edge. For human vision, the edge region has higher priority to embed the data than the smooth region. We propose the following function to compute the data hiding length of each coefficient W of each band as below:

$$L = \begin{cases} \mathrm{K}+3, \text{if } \mathrm{c} \ge 2^{K+3} \\ \mathrm{K}+2, \text{if } 2^{K+2} \le \mathrm{c} < 2^{K+3} \\ \mathrm{K}+1, \text{if } 2^{K+1} \le \mathrm{c} < 2^{K+2} \\ \mathrm{K}, \text{if } \mathrm{c} < 2^{K+1} \end{cases}, \quad 1 \le K \le 3 \tag{3}$$

where L is the length of data bits that are hided, K is the minimum length of each coefficient of each sub-band and c is the absolute coefficient value in the sub-band. According to equation (3), the coefficients are divided to four levels, level 1, level 2, level 3 and level 4. Higher levels are embedded more data, and lower levels are embedded fewer data. Suppose that the binary representation of the absolute value of the wavelet coefficient w is $a_7a_6a_5a_4a_3a_2a_1a_0$, and the hiding length $L = K + n, 0 \le n \le 3$. The bits $a_7a_6...a_L$ called unchangeable, since these bits cannot be changed. The bits $a_{L-1}a_{L-2}...a_0$ are called changeable. We replace these bits with secret data. Assume that a coefficient W in a sub-band is 9, and K = 1, the bitstream to be embedded in the coefficient is 010111000111011

Step 1: Since $2^{1+2} \le 9 < 2^{1+3}$, the hide length L = K + 2= 3

Step 2: According to Step 1 we can embed the first 3 bits 010_2 into the coefficient 9. Since the 8-bit binary representation of 9 is 00001001. We can just replace the 3 LSBs of 00001001by 010 to be 00001010, which is 10.

Since we want to hide the secret data in the image according to the complexity of the image content we analyze the complexity of two images called Lena and Baboon as shown in Fig. 2(a) and Fig. 2(b). For example, we compare the first 8*8 sub-block of Lena between the first 8*8 sub-blocks of Baboon. The parts are shown as Fig 2(c) and Fig 2(d). It is easy to understand that the part of Lena is smoother than the part of Baboon. We denote the first sub-block of Lena to ***A***, the first sub-block of Baboon to

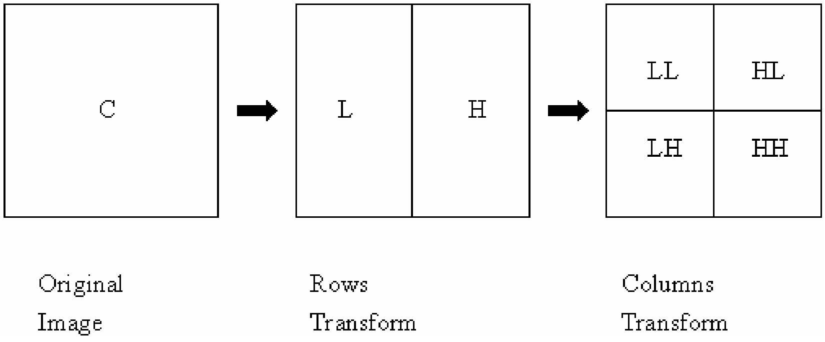
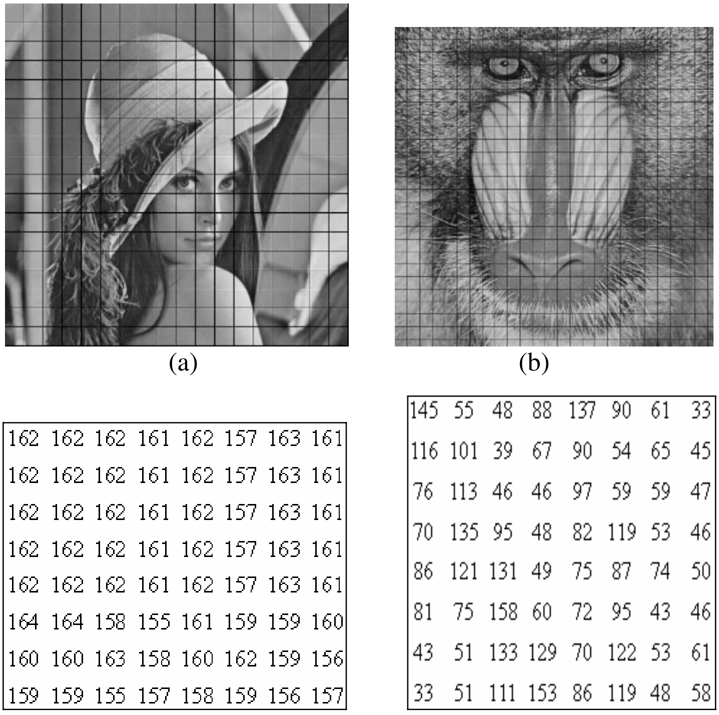


Fig. 1. 2D-HDWT decomposition



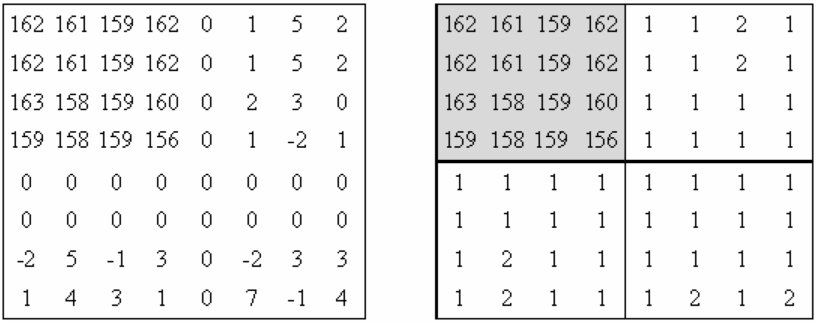
162	162	162	161	162	157	163	161
162	162	162	161	162	157	163	161
162	162	162	161	162	157	163	161
162	162	162	161	162	157	163	161
162	162	162	161	162	157	163	161
164	164	158	155	161	159	159	160
160	160	163	158	160	162	159	156
159	159	155	157	158	159	156	157

145	55	48	88	137	90	61	33
116	101	39	67	90	54	65	45
76	113	46	46	97	59	59	47
70	135	95	48	82	119	53	46
86	121	131	49	75	87	74	50
81	75	158	60	72	95	43	46
43	51	133	129	70	122	53	61
33	51	111	153	86	119	48	58

(c) (d)

Fig. 2. (a) and (b) are 8x8 blocks of Lena and Baboon ;(c) and (d) are the chosen block of Lena and Baboon

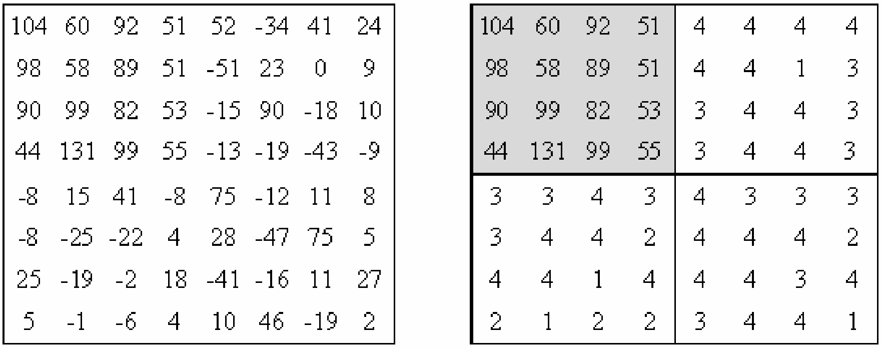
B. We de-compose ***A*** and ***B*** by 2D-HDWT, and use hiding capacity function with K = 1 to analyze the coefficients of A and B. We decompose ***A*** and ***B*** by 2D-HDWT, and use hiding capacity function with K = 1 to analyze the coefficients of A and B. Fig 3 and Fig 4 show the results of 2D-HDWT and the data hiding capacity for ***A*** and ***B***.



162	161	159	162	0	1	5	2
162	161	159	162	0	1	5	2
163	158	159	160	0	2	3	0
159	158	159	156	0	1	-2	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-2	5	-1	3	0	-2	3	3
1	4	3	1	0	7	-1	4

162	161	159	162	1	1	2	1
162	161	159	162	1	1	2	1
163	158	159	160	1	1	1	1
159	158	159	156	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	1
1	2	1	1	1	2	1	2

Fig. 3. (a) 2D-HDWT decomposition of A (b) data hiding of A



104	60	92	51	52	-34	41	24
98	58	89	51	-51	23	0	9
90	99	82	53	-15	90	-18	10
44	131	99	55	-13	-19	-43	-9
-8	15	41	-8	75	-12	11	8
-8	-25	-22	4	28	-47	75	5
25	-19	-2	18	-41	-16	11	27
5	-1	-6	4	10	46	-19	2

104	60	92	51	4	4	4	4
98	58	89	51	4	4	1	3
90	99	82	53	3	4	4	3
44	131	99	55	3	4	4	3
3	3	4	3	4	3	3	3
3	4	4	2	4	4	4	2
4	4	1	4	4	4	3	4
2	1	2	2	3	4	4	1

Fig. 4. (a) 2D-HDWT decomposition of B (b) data hiding capacity of B

We can find that the most coefficients of ***A*** fall into level 1; in terms of ***B***, the most coefficients fall into level 3 and level 4. Fig 5 shows 57.38% of the coefficients of all blocks of LENA embed 1-bits, while only 42.63% of the coefficients can embed more than 2-bits. In terms of Baboon, 22.75% of the coefficients embed 1-bits, and 77.25% of the coefficients can embed more than 2-bits. From the discussion above, our hiding capacity function only analyze the data hiding capacity of three bands except LL1 band. The block is determined to be an edge block with the quantity of level 1 and level 2 of these three bands fewer than 80% in total. Otherwise, it is a smooth block. The LL1 band is further decomposed in 4 sub bands if the 8* 8 block is an edge block. During the experiment, Lena contains 35% edge block while Baboon contains 93% edge blocks in 4096 blocks. We now propose our adaptive data hiding algorithm as below:

(1) Do histogram modification of the cover image.
(2) Divide the cover image into 8*8 block after histogram modification.
(3) Transform each block to frequency domain with 2D HDWT, and get four sub-bands LL1, HL1, LH1 and HH1.
(4) Decide the data hiding capacity L of each coefficient for HL1, LH1 and HH1 bands.

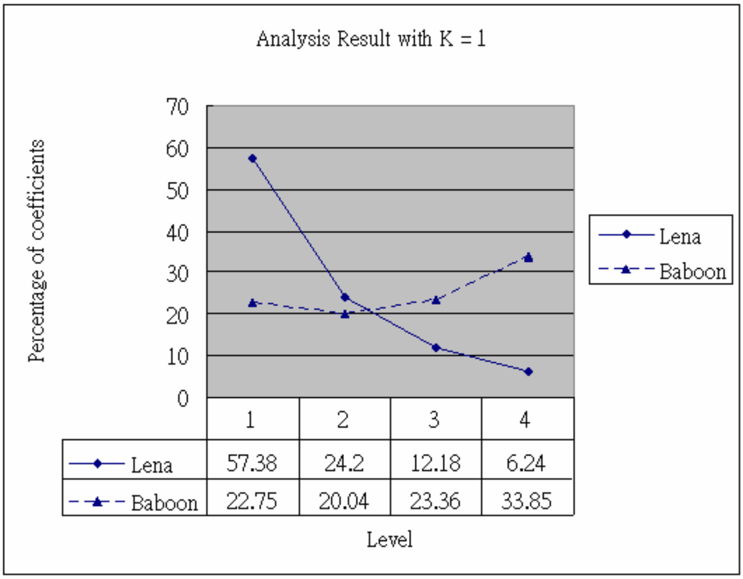


Fig. 5. Result by using Hiding capacity function to analysis the different image

(a) (b)

Fig. 6. (a) the magnification area is indicated by a square

(5) Represent each coefficient with sign magnitude, and embed L bits of data into the coefficient.

(6) Decide the hiding capacity of the LL1 band. If $\frac{Level3+Level4}{\text{Total}} \geq 0.2$ do 2D-HDWT of the LL1 band and decide the hiding capacity of HL2, LH2, HH2; otherwise do k-LSB on the LL1 band.

(7) Transform each 8 x 8 block into the spatial domain by inverse 2D-HDWT.

The extracting procedure is like the embedding procedure. When we get a stego-image, we divide it into 8*8 sub-blocks. We get four sub-bands after transforming

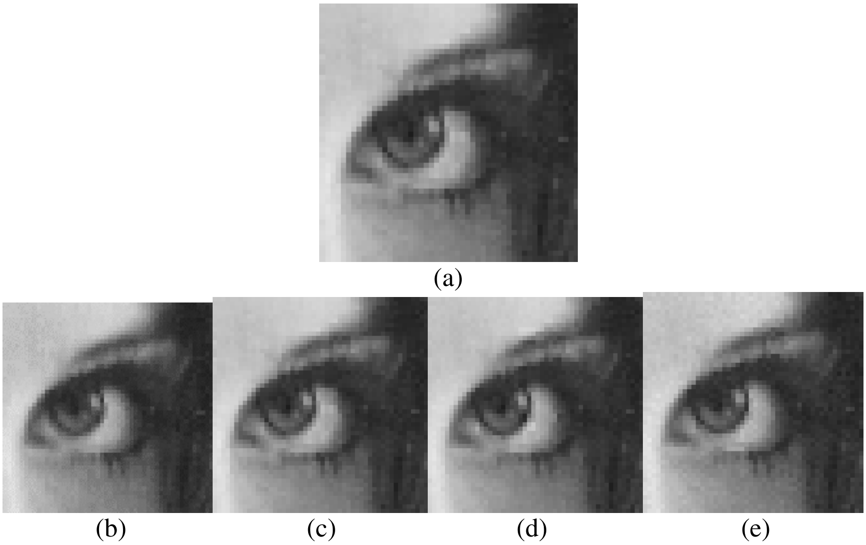

(a)
(b) (c) (d) (e)

Fig. 7. (a) original image (b) Wu's method (c) proposed method with K = 1 (d) K = 2 (e) K = 3

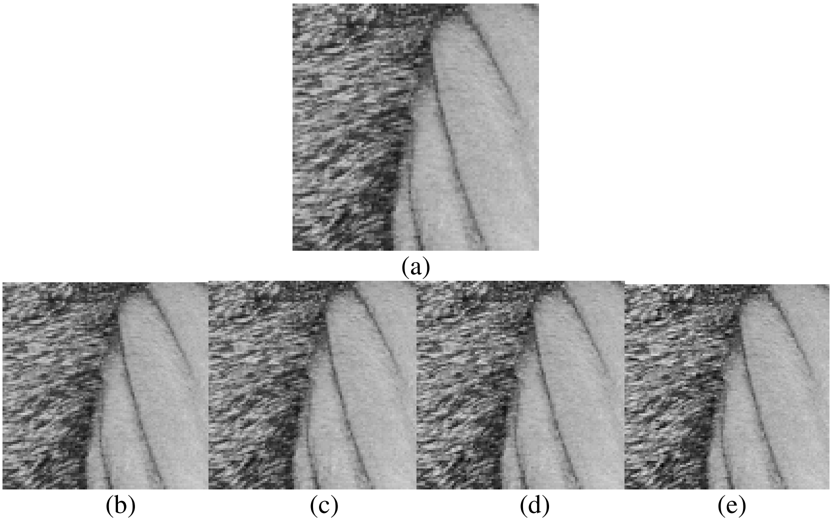
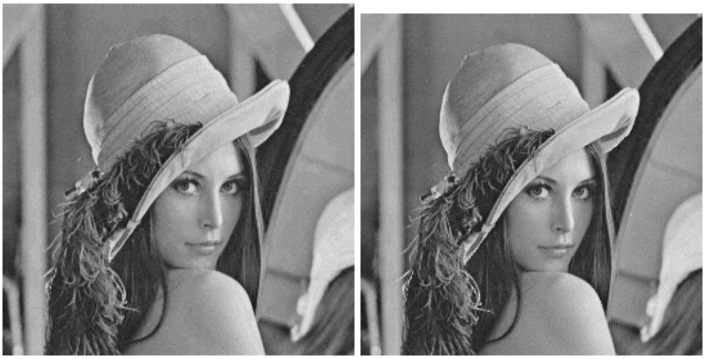

(a)
(b) (c) (d) (e)

Fig. 8. (a) original image (b) Wu's method (c)proposed method with K = 1 (d) K = 2 (e) K = 3

each block to wavelet domain. Because it has the same results of analyzing the cover image and the stego-image by hiding capacity function, we can use hiding capacity function directly to get the capacity of each coefficient of each band and extract the data from the coefficients.



(a) (b)

Fig. 9. (a) the worst case of LSB-3 (b) the worst case of our proposed method with K=3

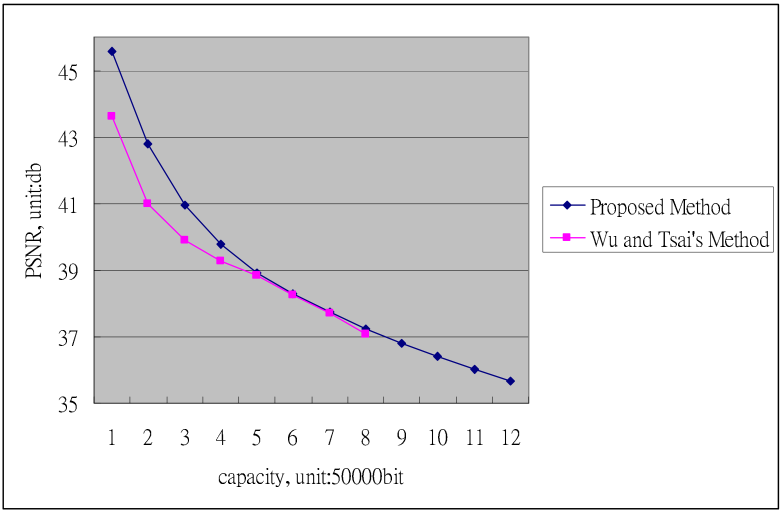
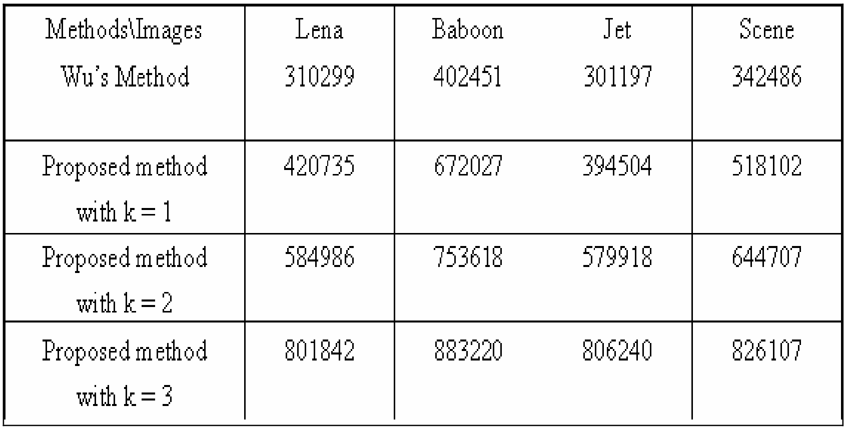


Fig. 10. Image quality and data capacity comparisons between Wu's and proposed method in Baboon

3 Experimental Results

In our experiment, we use two images called “Lena” and “Baboon”. Our experiments are executed on Intel Pentium 4 1.6GHz, 512 MB and Microsoft Windows XP SP2. The bitstream hidden in the image is generated by a random function. We compare our experimental results to other similar methods. In order to classify the difference of the experimental results, we take 50*50 blocks in Lena and Baboon and enlarge 5 times. The parts we chosen are shown as Fig 6; each segment contains smooth region and texture region. The enlarged sub-blocks are shown in Fig 7 and Fig 8. The results after embedding the worst data of simple LSB and our proposed method are shown in

Table 1. Comparison of capacity between our proposed method and Wu's method



Methods\Images	Lena	Baboon	Jet	Scene
Wu's Method	310299	402451	301197	342486
Proposed method with k = 1	420735	672027	394504	518102
Proposed method with k = 2	584986	753618	579918	644707
Proposed method with k = 3	801842	883220	806240	826107

Fig 9. It shows that our method outperforms Wu's method in data hiding capacity in visually same image quality. Fig. 10 shows that our method has higher PSNR that Wu's method in the same hiding bit rate. We compare the image quality under the same bit rate. Finally, the comparison results are shown in Table 1.

4 Conclusion

In this paper, we proposed a simple method of data hiding with Haar discrete wavelet transform. Since Human eyes are insensitive to the edge region we embed more data in the edge region. We use HDWT to embed secret data into each coefficient in LH, HL and HH bands. We define a "data hiding capacity function" to decide the data hiding capacity of the coefficient of these three bands. The experimental result shows that our capacity is better than other similar methods in both smooth images and texture images.

References

[1] Tewfik, A.H.; Swanson, M, "Data hiding for multimedia personalization, interaction, and protection" Signal Processing Magazine, IEEE, Vol. 14, No. 4,July 1997 pp. 41 – 44

[2] Barni, M.; Bartolini, F. " Data hiding for fighting piracy", Signal Processing Magazine, IEEE, Volume 21,No. 2,Mar, pp. 28 - 39

[3] Chi-Kwong Chan, L.M. Cheng, "Hiding data in images by simple LSB substitution", Pattern Recognition 37 (2004) 469 – 474

[4] Da-Chun Wu, Wen-Hsiang Tsai," A steganographic method for images by pixel-value differencing", Pattern Recognition Letters 24 (2003) 1613–1626

[5] A.Said and W.A.Pearlman, "An image multiresolution representation for lossless and lossy image compression", IEEE Trans. Image Process., vol.5 pp.1303-1330, Sep. 1996

[6] Guorong Xuan; Jidong Chen; Jiang Zhu; Shi, Y.Q.; Zhicheng Ni; Wei Su "Lossless data hiding based on integer wavelet transform" Multimedia Signal Processing, 2002 IEEE Workshop on, 9-11 Dec. 2002 Page(s): 312 – 315

Error Concealment in Video Communications by Informed Watermarking

Chowdary Adsumilli[1], Sanjit Mitra[2], Taesuk Oh[3], and Yong Cheol Kim[3]

[1] Citrix Co.
adsumilli@citrix.com
[2] Dept. of ECE, University of California, Santa Barbara
mitra@ece.ucsb.edu
[3] Dept. of ECE, University of Seoul, Korea
{bbole, yckim}@uos.ac.kr

Abstract. We propose an informed watermarking algorithm that aids in concealing packet loss errors in video communications. This watermark-based error concealment (WEC) method embeds a low resolution version of the video frame inside itself as watermark data. At the receiver, the extracted watermark is used as a reference for error concealment. The proposed DCT-based algorithm employs informed watermarking techniques to minimize the distortion of host frames. At the encoder, a predictive feedback loop is employed which helps to adjust the strength of the data embedding. Furthermore, the distortion of the DCT coefficients introduced in the embedding can be removed to a considerable extent, by employing bit-sign adaptivity. Simulation results on standard video sequences show that the proposed informed WEC scheme has an advantage of 3∼4 dB in PSNR over non-informed WEC.

Keywords: watermark, informed watermarking, error concealment, video communications.

1 Introduction

Packet losses that occur during the transmission of compressed video through lossy channels produce perceivable defects over multiple frames and have a significant influence on the quality of the received video at the end user. Error concealment is a technique that detects and hides the defects in video due to packet losses and therefore is one of the key processing steps in video communications. For the detection and correction of defects, error concealment techniques typically perform computationally intensive processing of spatial and temporal data in the received video, or they depend on certain critical information from the transmitter, like synchronization markers, to identify these packet loss errors.

Watermarking has been used for security applications where authentification and malicious attack prevention have been the primary focus. A new application for watermarking has been evolving lately for the error concealment in video communications. In this paper, we will refer to such schemes as watermark-based error concealment (WEC) schemes. In this work, we propose an efficient

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1094–1102, 2006.

WEC approach where a low resolution version of a video frame is embedded in itself at the encoder in the form of watermark data.

The low resolution image is a binary image with 1/16-th the size of the original frame. It is obtained by halftoning a second level 2D-DWT of each frame. The low resolution image is embedded in 2×2 format into the mid-frequency coefficients of a full-frame DCT of the original frame. At the receiver, a correlational detector is used to extract the low resolution image, which is used as a reference to identify and conceal any packet loss errors. We validate the use of full-frame DCT by comparing it with the block-based DCT embedding scheme.

The algorithm used in this paper has some features of informed watermarking in that the encoder makes an informed decision in the embedding process. This informed embedding makes it possible to minimize the distortion of the host frames. At the encoder, a predictive feedback loop is employed which estimates the watermark detection accuracy at the receiver. Then, the strength of the scale factor α is determined such that BER at the decoder stays under some threshold. Furthermore, some of the modified coefficients of the DCT signal are virtually free from distortion by employing bit-sign adaptivity.

The organization of this paper is as follows. Previous works on WEC are presented in Chapter 2. The processes of embedding and extracting the watermarking for error concealment are described in Chapter 3 and details of the in formed watermarking are in Chapter 4. Simulation results on test sequences are presented in Chapter 5. Finally, we draw a conclusion in Chapter 6.

2 Previous Works

The concept of applying watermarking to error resilience and concealment applications is quite new. Typically, as a simple form of WEC for image data, certain key features were extracted from the image, encoded, and hidden in the image itself either as a resilience tool or for concealment. The concept was extended to video coding by Bartolini *et. al.*[3]. However, they used data hiding as a tool to increase the syntax-based error detection rate in H.263, but not for the purpose of recovering lost data.

Munadi *et. al.* extended the concept of key feature extraction and embedding to inter-frame coding [4]. In their scheme, important features are embedded into the prediction error of the current frame. However, the effects on motion vectors and the loss of motion compensated errors were not addressed. Yilmaz [5] proposed embedding a combination of edge information, block bit-length, and parity bits into intra-frames. They use even-odd signaling of DCT coefficients for embedding, which is minimally robust.

Informed watermarking has also been studied as a part of watermarking research [6]. A number of concealment techniques that do not use watermarking while giving similar high levels of performance have also been proposed [7]. However, WEC methods proved to give an improvement over the other techniques. A comparison of these techniques with the proposed one is provided in [1].

The problems with existing techniques are that (1) only one or a few selected set of key features are used for embedding. These features may not necessarily follow the loss characteristics of the channel employed and so are not effective for error resilience, and (2) they often use frequency domain such as DCT to encode the key data to be embedded. However, if losses occur on the DC or a set of the first few AC coefficients, the loss to the extracted reference would be significant and therefore may lead to reduction in concealment performance. Our proposed technique avoids these problems by embedding a halftoned low resolution version of the whole reference image. This way, loss or errors in the data will have smaller and local effects on the reconstructed video.

3 Watermark-Based Error Concealment

3.1 Watermark Embedding

The watermarking scheme used in this paper is a modified version of the Cox's algorithm [8]. In this work, Harr discrete wavelet transform (DWT) and halftoning techniques are used to generate the low resolution image, which is the watermark data to be embedded. This way, the DWT approximation image is transformed into binary values before being embedded. The block diagram of the embedding algorithm is shown in Figure 1. A second level 2D-DWT is performed on the video frame to obtain an image that is 1/16-th the size of the original frame. A halftoned image, the marker, is then generated from the reduced image. One marker is used for each key frame in the video. Each pixel of the marker is embedded 4 times in a 2 × 2 matrix format. Details are as follows.

Let the k-th frame be f_k with $m \times n$ pixels. The 2nd level DWT approximation image, α_k has a size of $m/4 \times n/4$, which is halftoned into a binary-valued marker, m_k, with the same size of $m/4 \times n/4$. The dithering technique in the halftoning is Floyd-Steinberg error diffusion algorithm [9]. Each pixel of the marker is then repeated in a 2×2 format to form w_k, which has a size $m/2 \times n/2$. Then, the final watermark data, $\tilde{w}_k$, is generated as follows ($\cdot *$ represents element-by-element multiplication). p_k is a $m/2 \times n/2$-sized array of zero-mean unit-variance Gaussian.

$$\tilde{w}_k = w_k \cdot * p_k \qquad\qquad w_k(i,j) \in \{-1, +1\}$$

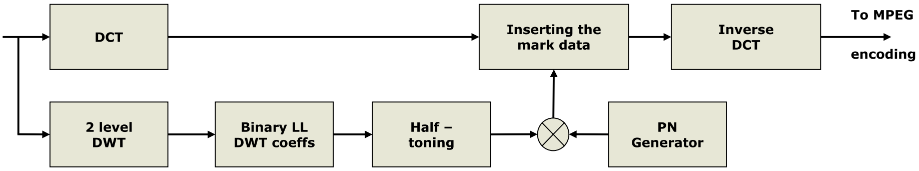


Fig. 1. Block diagram of the embedding algorithm

Let x_k be the DCT coefficients of the luminance channel of the frame. The watermark, $\tilde{w}_k$ is then scaled by a factor, α, and then added to a mid-frequency set of coefficients, starting at the initial frequencies of (Δ_1, Δ_2). The modified DCT coefficients y_k are obtained by

$$y_k(i+\Delta_1, j+\Delta_2) = x_k(i+\Delta_1, j+\Delta_2) + \alpha \cdot \tilde{w}_k(i,j) \tag{1}$$

$$\begin{pmatrix} 0 \leq i < m/2, & 0 \leq j < n/2 \\ \Delta_1 \in [0, m/2], & \Delta_2 \in [0, n/2] \end{pmatrix}$$

where i and j correspond to the pixel location in the watermark data and also the DCT coefficients. Finally, y_k is inverse transformed, encoded in a compressed form and then transmitted.

3.2 Watermark Extraction

The extraction of the watermark employs a correlational receiver as shown in Figure 2. The received (noisy) DCT coefficients of the luminance channel, $\tilde{y}_k$ are multiplied by the same pseudonoise array p_k and then summed for each 2×2 block. Then the binary marker is extracted by taking the polarity of the sum.

$$\hat{m}_k(i,j) = \begin{cases} 1, & \text{if} \quad \lambda_k(i,j) \geq 0 \\ 0, & \text{if} \quad \lambda_k(i,j) < 0 \end{cases}$$

$$\lambda_k(i,j) = \sum_{\substack{\acute{i}=2i-1, \\ \acute{j}=2j-1}}^{2i,2j} \tilde{y}_k(\acute{i}+\Delta_1, \acute{j}+\Delta_2) \cdot p_k(\acute{i}, \acute{j}) \quad (0 \leq i < m/4, 0 \leq j < n/4) \tag{2}$$

An inverse halftoning algorithm proposed by Xiong [10] is applied to $\hat{m}_k$ to obtain an estimate of the low-resolution approximation image, $\hat{a}_k$. A 2-D inverse DWT is performed on $\hat{a}_k$ to obtain an intermediate resolution image, b_k $(m/2 \times n/2)$. Then, b_k is zoomed by a factor of 2 by up-sampling and passing through a lowpass filter to obtain a reference image, g_k $(m \times n$ size). An estimate of the current frame, $\hat{f}_k$ is obtained by decompressing the received video data packets. The reference g_k is compared with $\hat{f}_k$ to detect and conceal the corrupted areas of $\hat{f}_k$. More details on WEC operations can be found in [1] and [2].

4 Informed Watermarking Algorithm

Data embedding in this work is based on informed watermarking. The strength of the embedded watermark varies adaptively by employing a predictive feedback loop. This adaptivity highly decreases not only the BER of the extracted watermark but also the perceivable distortion in the video introduced by the watermarking process. Two variations of the informed methods are explained in detail herein.

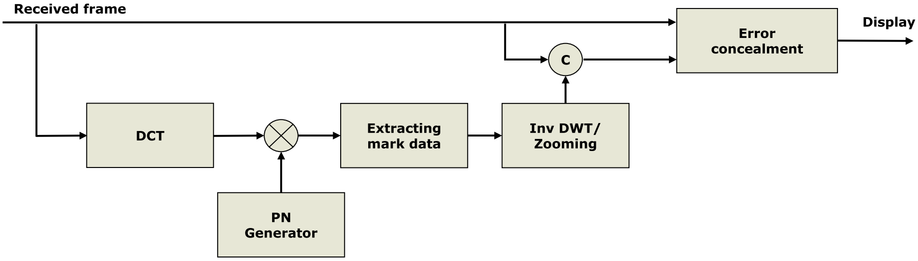


Fig. 2. Block diagram of the retrieval algorithm

4.1 Predicted Watermark Detection

The embedder has a predictive detector which is connected in a feedback loop to adapt to the strength of the embedding host signal, as shown in Figure 3. This allows the values of the scale factor α to vary such that the probability of error in detecting the watermark at the receiver is minimized. Starting from a small fixed value, α continues to vary in incremental steps until the watermarked bit is extracted correctly in the predictive detector for a target level of packet errors. When it is correctly extracted, the value of α is fixed for that coefficient and therefore (1) can be restated as follows.

$$y_k(i+\Delta_1, j+\Delta_2) = x_k(i+\Delta_1, j+\Delta_2) + \alpha_k(i,j) \cdot \tilde{w}_k(i,j) \tag{3}$$

The advantages of this informed WEC method over the other WEC methods discussed in Section 3 are threefold: (1) the perceivable watermarking defects are negligible due to the adaptive scaling of the strength of the embedded bits, (2) the BER values are much lower in the informed technique when compared to the non-informed WEC methods, and (3) a higher level of compression can be achieved at the codec as the entropy of the embedded video frame is not as high in case of the informed WEC technique since smaller α implies a reduction in the modification of the host coefficients.

4.2 Watermark Bit-Sign Adaptivity

In this scheme, the watermark bit is embedded by adapting the host frame coefficient with the sign of the pseudo-random number. A data bit is embedded by modifying the sign of the coefficient in accordance with the sign of x_k and p_k. This could be considered as a form of multiplicative embedding.

$$\begin{aligned} y_k(i+\Delta_1, j+\Delta_2) &= x_k(i+\Delta_1, j+\Delta_2) \cdot (\alpha_k(i,j) \cdot \tilde{w}_k(i,j)) \\ &= x_k(i+\Delta_1, j+\Delta_2) \cdot (\alpha_k(i,j) \cdot m_k \cdot p_k(i,j)) \end{aligned} \tag{4}$$

As before, one watermark bit, $m_k = -1$ or $+1$, is embedded into four DCT coefficients. When $m_k = +1$, the sign of y_k is equal to that of the product of

x_k and p_k, otherwise, it is the opposite of $x_k \cdot p_k$. At the detector, the value of the watermark bit is extracted in a correlational receiver similar to (2) from the lossy received signal, $\tilde{y}_k$.

$$\hat{m}_k(i,j) = \begin{cases} 1, & \text{if} \quad \lambda_k(i,j) \geq 0 \\ 0, & \text{if} \quad \lambda_k(i,j) < 0 \end{cases}$$

$$\lambda_k(i,j) = \sum_{\substack{\acute{i}=2i-1, \\ \acute{j}=2j-1}}^{2i,2j} \tilde{y}_k(\acute{i}+\mathit{\Delta}_1, \acute{j}+\mathit{\Delta}_2) \cdot p_k(\acute{i},\acute{j}) \quad (0 \leq i < m/4, 0 \leq j < n/4) \quad (5)$$

Figure 4 shows the use of informed watermarking in bit-sign adaptivity. When α is fixed, λ_k does not necessarily reproduce the data bit and so the original data bits of $(+1, -1)$ may be erroneously decoded into $(-1, -1)$ even in the absence of channel noise. On the other hand, the value of α can be adjusted from the feedback in informed watermarking such that the data bit is correctly recovered in the predictive detector at the transmitter. With this bit-sign adaptivity, the original data bits are always correctly reproduced if there is no packet error.

This technique has additional advantageous features: It reduces the host frame distortion: Once the watermark is correctly detected, the coefficient in the host frame can return to some value with its original polarity by simply multiplying the pseudonoise matrix with the embedded coefficients.

$$\begin{aligned} \hat{x}_k(i+\mathit{\Delta}_1, j+\mathit{\Delta}_2) &= y_k(i+\mathit{\Delta}_1, j+\mathit{\Delta}_2) \cdot p_k(i,j) \\ &= [\hat{m}_k(i,j) \cdot \alpha_k(i,j) \cdot p_k(i,j)^2] \cdot x_k(i+\mathit{\Delta}_1, j+\mathit{\Delta}_2) \quad (6) \end{aligned}$$

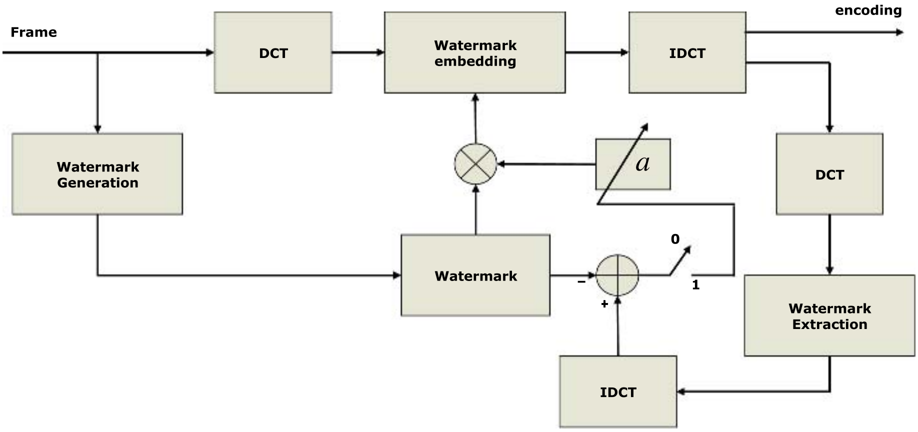


Fig. 3. Feedback-based watermark embedding model

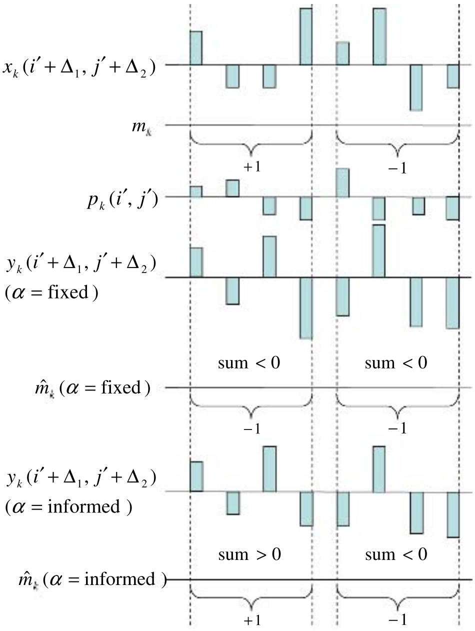
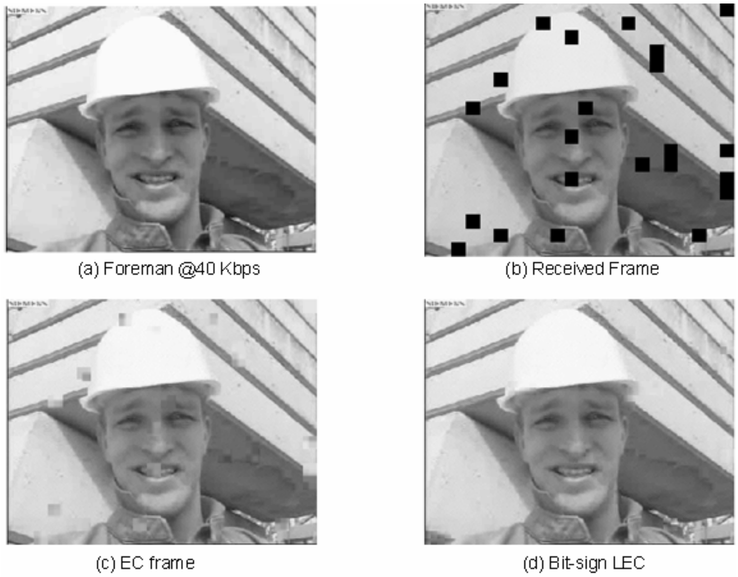

$x_k(i'+\Delta_1, j'+\Delta_2)$
m_k
+1
−1
$p_k(i', j')$
$y_k(i'+\Delta_1, j'+\Delta_2)$
(α = fixed)
sum < 0
sum < 0
$\hat{m}_k$ (α = fixed)
−1
−1
$y_k(i'+\Delta_1, j'+\Delta_2)$
(α = informed)
sum > 0
sum < 0
$\hat{m}_k$ (α = informed)
+1
−1

Fig. 4. Data bits are correctly recoveed with bit-sign adaptivity


(a) Foreman @40 Kbps

(b) Received Frame

(c) EC frame

(d) Bit-sign LEC

Fig. 5. The results for the 36-th frame of Foreman in CIF resolution compressed to 40 Kbps. The PSNR values are (b)14.26 dB (c) 27.63 dB (d)36.25 dB, respectively.

5 Experimental Results by Simulations

A sample set of CIF resolution (288 × 352) videos is considered for simulation and the watermark is inserted in the central AC frequencies of the full frame DCT. NS-2 simulator is used to generate packet losses with a two-state Gilbert-Elliot Gaussian packet loss model with predefined mean and variance. The packet size was fixed at a macro block size and no retransmissions were allowed.

Figure 5 shows the performance of WEC on the *Foreman* video sequence. The received frame is obtained for a mean packet loss probability of 0.12% and variance 0.02%. The value of α was fixed at 0.6 for the non-informed WEC case. Figure 5(b) shows the lossy received frame while Figure 5(c) shows the error concealed (EC) frame. Since the watermark is a low resolution version, most of the high frequency information is not retained. Therefore, the reference frame looks *smooth* and the EC frame is *patchy.* This effect is reduced if the EC frame is locally-scaled based on the neighboring luminance values. We call the resulting frame as locally-scaled error concealed (LEC) frame. Figure 5(d) represents the LEC frame combined with bit-sign adaptivity.

Table 1 shows the performance of the proposed algorithms over various video sequences. As seen, for all cases except *Flower* and *Highway*, the bit-sign adaptive informed WEC gave higher PSNR values of about 3 dB over the non-informed WEC technique.

Table 1. PSNR (in dB) for a fixed mean loss 0.15% and variance 0.025%

		Non-informed WEC		Informed WEC		
Video	Received	EC	LEC	EC	LEC	BitSign
Akiyo	16.1248	28.9114	31.5345	31.9824	34.7662	35.5219
Foreman	14.8279	26.6248	31.4983	32.6751	34.3655	35.5787
Table tennis	15.8202	27.4553	30.1257	31.6767	32.9044	35.0286
Flower	14.5233	26.4553	30.0476	30.8902	32.0967	33.3261
Football	15.0728	27.3546	30.5126	31.0207	33.2548	35.3652
Paris	14.6905	27.0817	30.9213	31.0576	32.9247	33.6563
Highway	15.0253	28.7346	32.4018	32.9790	34.3862	34.2892

6 Conclusions

We presented an application of informed watermarking algorithm to video error concealment. This WEC approach used full-frame DCT to embed a low resolution version of the video frame in itself. The extracted watermark was used for error concealing in the lossy received video frame. The algorithm employed a feedback loop to predict the values of the extracted watermark bits, thereby reducing the overall BER of detected watermark at the receiver. Bit-sign coefficient modifications were also presented.

Based on the obtained results, we conclude that the informed watermarking algorithm gave better performance not only in terms of higher PSNR values but also in terms of reduced BER values. The bit-sign variation proved to reduce the perceivable defects introduced by watermarking process, when compared to the non-informed WEC method.

Acknowledgment. This work was supported in part by the Brain Korea 21 of DOE, Korea, in part by National Science Foundation grant no. CCR-0105404, in part by University of California MICRO grant with matching supports from Philips Research Laboratories and Microsoft Corporation.

References

1. Adsumilli, C.B., Farias, M.C., Mitra, S.K., Carli, M, " A robust error concealment technique using data hiding for image and video transmission over lossy channels," Vol. 15, No. 11, IEEE Trans. Circuits and Systems for Video Technology , Nov. 2005 pp.1394 - 1406
2. Adsumilli, C.B.; Mitra, S.K.; Kim, Y.C., "Detector Performance Analysis of Watermark-Based Error Concealment in Image Communications," Proc. of ICIP, IEEE, Sept. 2005 pp.:916 - 919
3. Bartolini, F. et al. A data hiding approach for correcting errors in H.263 video transmitted over a noisy channel. Proc. IEEE Workshop on Multimedia Signal Processing (2001) pp. 65-70
4. Munadi, K., Kurosaki, M., Kiya, H., " Error concealment using digital watermarking technique for interframe video coding," Proc. Intl. Tech. Conf. on Circuits and Systems, Computers, and Communications (2002) pp. 599-602
5. Yilmazi, A., Alatan, A., "Error concealment of video sequences by data hiding," Proc. of ICIP, IEEE, 2003, pp.679-682
6. Miller, M.L., Cox, I.J., Bloom, J.A., "Informed embedding: Exploiting image and detector information during watermark insertion," Proc. of ICIP, IEEE, 2000
7. Salama, P. et al, "Error concealment techniques for encoded video streams," Proc. ICIP, IEEE, 1995, pp.9-12
8. Cox, I., Kilian, J., Leighton, F., Shamoon, T., "Secure spread spectrum watermarking for multimedia," IEEE Trans. Image Processing 6(12) (1997) pp.1673-1687
9. Floyd, R.W., Steinberg, L., "An adpative algorithm for spatial gray-scale," Proc. Society of Information Display 17(2) (1976) pp.75-78
10. Xiong, Z., Orchird, M.T., Ramachandrani, K., "Inverse half-toning using wavelets," IEEE Trans. Image Processing 8(10) (1999) pp.1479-1483

Dirty-Paper Trellis-Code Watermarking with Orthogonal Arcs

Taekyung Kim, Taesuk Oh, Yong Cheol Kim, and Seongjong Choi

Department of Electrical and Computer Eng, University of Seoul, Korea
{ktk1010, bbole, yckim, chois}@uos.ac.kr

Abstract. Dirty-paper trellis-code watermarking with random-valued arcs is slow since the embedding into the cover work is performed on path-level which entails many Viterbi decodings through random convergence. We present a fast deterministic embedding in a trellis-code with orthogonal arcs. The proposed algorithm has a speedup factor of the message size since it is based on arc-level modification of the cover work in a bit-by-bit manner. Experimental results show that the proposed embedding provides higher fidelity and lower BER against various types of attacks, compared with conventional informed methods.

Keywords: informed watermarking, trellis-code, orthogonal arc.

1 Introduction

Digital watermarking is analogous to digital transmission. From the detector's viewpoint, the data is the watermark and the cover work is the channel noise. Since the dirty-paper channel model [1] showed that Gaussian noise known at the transmitter side does not degrade the channel capacity, several informed watermarking schemes have been developed [2]-[6], which adapts the watermarking signal to the cover work. Informed watermarking has superior performance, compared to watermarking with blind embedding. Without intentional attack, it can achieve a zero error probability in message extraction.

Informed watermarking usually consists of informed coding and informed embedding. In Figure 1, the message, $\boldsymbol{m}$ is encoded into $\boldsymbol{w_m}$. Then $\boldsymbol{w_m}$ is modified into $\boldsymbol{w_a}$. Finally, $\boldsymbol{w_a}$ is added to the cover work, $\boldsymbol{c_0}$ into $\boldsymbol{c_w}$. In informed coding, $\boldsymbol{m}$ is represented by several codewords which are dependent on $\boldsymbol{c_0}$. We choose the codeword $\boldsymbol{w_m}$ which will cause the least distortion to $\boldsymbol{c_0}$. In informed embedding, the embedder can freely decide which would be the final watermarked image. Hence, it can select any image as $\boldsymbol{c_w}$ by letting $\boldsymbol{w_a}$= $\boldsymbol{c_w}$ - $\boldsymbol{c_0}$. The watermarked image $\boldsymbol{c_w}$ should be perceptually close enough to $\boldsymbol{c_0}$ and be robust enough against malicious attacks.

Quantization index modulation (QIM) [2] is a lattice-based code which employs a family of indexed quantizers optimized for the cover work. QIM accommodates a large payload with fast encoding and decoding. A problem with QIM is its vulnerability to valuemetric attack, *i.e.* global intensity scaling.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1103–1112, 2006.

Trellis-code watermarking by Miller [3] can accommodate a large payload and is robust against valuemetric attack. Since message decoding is based on accumulated correlation, trellis-code is inherently robust against valumetric attack. Though the performance is excellent, a problem lies with the random-valued arcs in the trellis. Since the informed embedding is based on an optimization over random arc paths, it has a complex structure and requires a huge computation. Lin [7] proposed a faster embedding which uses pre-computed tables in Viterbi decoding. However, embedding a new message requires no less computation.

The dirty-paper watermarking by Abrardo [8] employs orthogonal equi-energetic codewords. Orthogonality among codewords provides a fast convergence in the embedding process. Though the performance from simulation on turbo-structured multi-stage decoding is excellent, the performance tested on real images is outperformd by the numerical simulation [9].

In this paper, we present an informed embedding in a dirty-paper trellis-code with orthogonal arcs. The proposed algorithm is combining the advantages of Miller's trellis-code [3] and Abrardo's orthogonal codewords [9] into one. As a consequence, our method has advantages over both Miller's and Abrardo's.

The embedding process in [3] is very slow since the modification of the cover work is performed on path level and Viberbi decoding is repeated over the whole path each time a single bit is embedded. Our method is faster by a factor of the message size, due to the arc-level embedding. Modification of the cover work needs only the incremental correlation of an arc, stage by stage.

Since Abrardo's method depends on the complex-structured turbo-decoding for flat robustness over the message bits and thus suffers a loss in the data capacity, the proposed algorithm makes use of the inherent robustness of a trellis-structured convolutional code and so there is no loss in data capacity.

This paper is organized as follows: In Section 2, the framework for the proposed method is described. In Section 3, informed coding and its comparison with [3] are described in details. In Section 4, experimental results are presented and we draw a conclusion in Section 5.

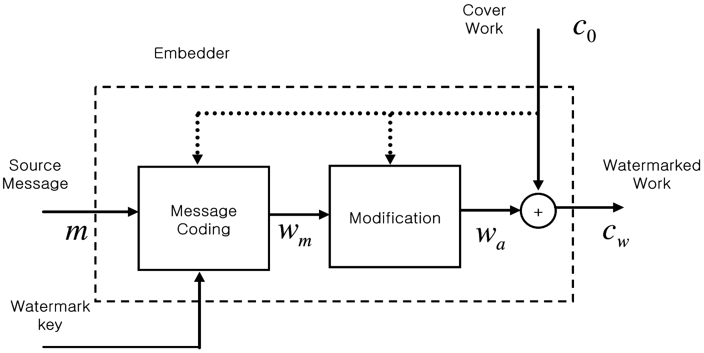


Fig. 1. Informed watermarking and embedding

2 Proposed Orthogonal Trellis-Code

The proposed dirty-paper trellis-code watermarking accommodates L bits of message in an L-staged trellis. The overall structure of the trellis and the informed coding part are similar to [3]. What we improved is the use of orthogonal arcs and the embedding algorithm which performs very efficiently over the orthogonal arcs. Details of the proposed method follow and a comparison with [3] is presented for the parts where they are different.

2.1 Trellis-Code with Orthogonal Arcs

The trellis-code used in our experiments is shown in Figure 2. The arcs used in the trellis are the column vectors in an $N \times N$ Hadamard matrix. Hence, the number of available orthogonal arcs is N, in all. An arc $\mathbf{h_i}$ represents the i-th column of the matrix.

The assignment of arcs to nodes may have different configuration from stage to stage. For efficiency of implementation, we use identical assignment of arcs to nodes for all stages. The performance is not influenced by a permutation of the arcs. Though the number of all the orthogonal arcs is limited to N, the number of arcs ($= M$) between two layers may be different from N. All the arcs are distinct when $M \leq N$, and some arcs are linear combinations of the N basis vectors when $M > N$. In this experiment, we set M to be equal to N.

In each node in the trellis, there are $M/4$ incoming arcs and $M/4$ outgoing arcs. Bold arcs represent positive arcs which embed bit 1 and non-bold arcs represent negative arcs which represent bit 0. $\mathbf{H}^*$ denotes the set of all the positive and negative arcs.

$$\mathbf{H}^* = \{\mathbf{h_1}, \mathbf{h_2}, \cdots, \mathbf{h_M}\}$$

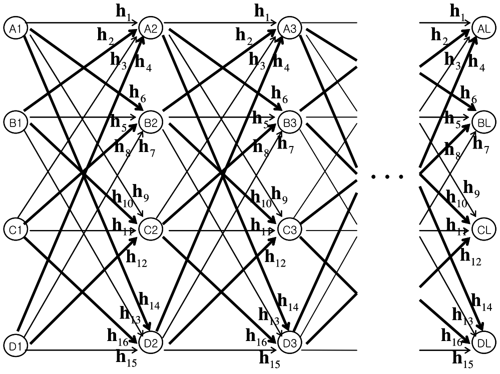


Fig. 2. Trellis-code with orthogonal arcs

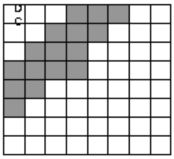


Fig. 3. The 12 DCT coefficients for the extract vector

2.2 Embedding and Detection of Message Bits

The image is converted into 8 × 8 block DCT domain. Then, N low ac components are taken from each of the L blocks, as in Figure 3, and then they are concatenated and shuffled into a randomly-ordered vector. The shuffled vector, $\mathbf{v}$, referred as the extract vector of the image, is an $L \times N$ sequence. The i-th message bit is embedded into the i-th N-element segment in $\mathbf{v}$.

$$\mathbf{v} = [v_{1,1}, v_{1,2} \cdot \cdot\cdot, v_{1,N} \cdots v_{i,1}, v_{i,2} \cdots v_{i,N} \cdots v_{L,1}, v_{L,2} \cdots v_{L,N}] \tag{1}$$

3 Informed Embedding over Orthogonal Arcs

Informed coding is the process of choosing the codeword which is the most correlated with the cover work. In the proposed trellis, there are 4^L paths in all, starting from the node in the first stage to the end, since 4 nodes exit from each node. Each path represents a specific (but not necessarily distinct) message.

Identification of the right path is performed in two steps. In the first step, we remove those arcs in the trellis which do not encode the given message. Then, each stage is left with either bold arcs (bit 1) or non-bold arcs (bit 0). All the paths in the modified trellis shown in Figure 4 represent the encoded message. In the second step, the highest correlation path is identified by Viterbi decoding on the modified trellis.

At the receiver side, Viterbi decoding on the complete trellis might return a path other than the path chosen for the modified trellis. To suppress this erroneous decoding, the extract vector is modified such that its correlation with the arcs in the right path secures a margin over all the interfering paths.

3.1 Informed Embedding

The description that follows is about the embedding process for the i-th stage in the trellis. However, it can be equally applied to all the other stages since they have identical configuration of arcs. Without loss of generality, we may assume that the i-th message bit is 1 and the arc in the i-th stage in the identified path from the informed coding process is $\mathbf{h_{m0}}$.

Define $\mathbf{u}$ be the i-th N-element segment in $\mathbf{v}$.

$$\mathbf{u} = [u_1, u_2 \cdots u_N] = [v_{i,1}, v_{i,2} \cdots v_{i,N}]$$

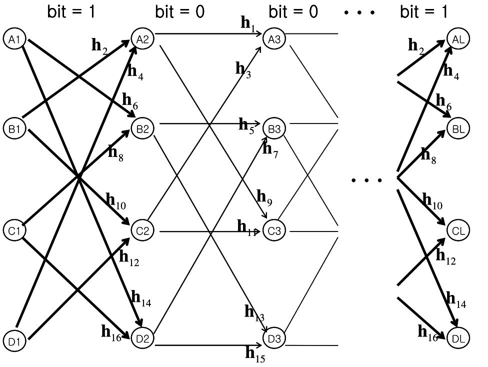


Fig. 4. Modified Trellis-code with only unipolar arcs

Define q_m as the step correlation of an arc, $\mathbf{h_m}$ with $\mathbf{u}$.

$$q_m = \mathbf{h_m} \cdot \mathbf{u} \qquad (m = 1, 2, \cdots, M) \tag{2}$$

Viterbi algorithm returns the path which wins the highest accumulated correlation with $\mathbf{v}$. The accumulated correlation is computed incrementally for each stage of the trellis. We define $Q^i_m(j)$ where $(j = 1, ..., i)$ as the sum of correlation of the arcs with $\mathbf{v}$. The arcs used in sum are only those starting from first stage to the j-th stage, out of all the i arcs in the winning path connected to $\mathbf{h_m}$ in the i-th stage.

$$Q^i_m(i) = q_m + Q^i_m(i-1) \tag{3}$$

When $\mathbf{h_{m0}}$ was chosen in the informed coding process, there were only positive arcs in the i-th stage of the modified trellis. Hence, there is always a chance of some negative arcs having larger accumulated correlation than $\mathbf{h_{m0}}$, and so the detector at the receiver side might return an erroneous decoded message since the detection is performed on the complete trellis. In order to prevent this, we modify $\mathbf{u}$ such that $Q^i_{m0}(i)$ is larger than the accumulated correlation for the other positive arcs as well as all the negative arcs, by at least R_0 (desired robustness constant).

$$Q^i_{m0}(i) \geq Q^i_m(i) + R_0 \qquad (m = 1, \cdots, M \quad m \neq m_0) \tag{4}$$

The N-element vector, $\mathbf{u}$, is updated element by element. Each element of $\mathbf{u}$ can be either increased or decreased in accordance with the signs of the corresponding elements in $\mathbf{u}$, $\mathbf{h_{m0}}$ and interfering arcs, $\mathbf{h_m}$.

When the signs of $\mathbf{u}$ and $\mathbf{h_{m0}}$ are the same and the sign of an interfering arc, $\mathbf{h_c}$ is different, we increase that magnitude of the element of $\mathbf{u}$ by α. When the signs of $\mathbf{u}$ and $\mathbf{h_c}$ are the same and the sign of $\mathbf{h_{m0}}$ is different, we decrease that magnitude of that element of $\mathbf{u}$ by α. If the element of $\mathbf{u}$ is too small, we increase or decrease it by β, after checking the signs in $\mathbf{h_{m0}}$ and $\mathbf{u}$. Modification of $\mathbf{u}$ is repeated against all the other interfering arcs.

As a result, each element of $\mathbf{u}$ is altered only when it helps to increase $Q^i_{m0}(i)$ over some of the interfering arcs. Furthermore, the amount of increase of

$Q^i_m(i)(m \neq m_0)$ at the change of one element of $\mathbf{u}$, never exceeds the increase in $Q^i_{m0}(i)$. With this updating scheme, the correlation margin over interfering arcs either increases or remains the same, at each element change of $\mathbf{u}$. This way, the proposed algorithm effectively modifies $\mathbf{u}$ toward $\mathbf{h_{m0}}$ and suppresses the interfering arcs until a desired robustness level of R_0 is achieved. The pseudo-code for the proposed embedding algorithm is as follows:

```
DO from $m = 1$ to $M$ (skipping $m \neq m_0$)
  $\mathbf{h_c} \leftarrow$ $m$-th element of $\mathbf{H}^*$
  REPEAT UNTIL ($Q^i_{m0}(i) \geq Q^i_m(i) + R_0$)
    Do from $n = 1$ to $N$
      $u \leftarrow n$-th element of $\mathbf{u}$
      $g \leftarrow n$-th element of $\mathbf{h_{m0}}$
      $b \leftarrow n$-th element of $\mathbf{h_c}$
      IF $u \cdot g \approx 0$
       THEN
       IF $g > 0$
         $|u| \leftarrow |u| + \beta$
       ELSE IF $g < 0$
         $|u| \leftarrow |u| - \beta$
       END IF
     END IF
     IF $u \cdot g > 0$   AND   $g \cdot b < 0$
     THEN
      $|u| \leftarrow |u| \cdot (1 + \alpha)$
     END IF
     IF $u \cdot g < 0$   AND   $g \cdot b < 0$
     THEN
      $|u| \leftarrow |u| \cdot (1 - \alpha)$
     END IF
    END DO
  END REPEAT
END DO
```

3.2 Comparison with Miller's Embedding

In the informed embedding in Miller's [3], the modification is performed on $\mathbf{v}$, instead of $\mathbf{u}$, in a stochastic manner. Each time $\mathbf{v}$ is updated, the accumulated correlation of interfering paths (rather than interfering arcs) with $\mathbf{v}$ also changes in a random way. This introduces two problems.

First, at each updating of $\mathbf{v}$, Viterbi decoding for all the 2^L paths needs to be performed again in order to find the most interfering path and secure the desired robustness against it. Even with the modified (simplified) Viterbi algorithm, it costs a huge computation time. Second, the modification of $\mathbf{v}$ is accomplished in a random way and so the convergence is not guaranteed.

The proposed embedding algorithm is free from these problems. First, convergence of the extract vector is deterministic since the updating of $\mathbf{u}$ is performed in accordance with the signs of $\mathbf{u}$ and arcs. At each updating of $\mathbf{u}$, it gets closer to $\mathbf{h_{m0}}$ and recedes farther from interfering arcs. Second, the embedding process is performed on $\mathbf{u}$, instead of along a path, which is L-times longer. Hence, just incremental updating of the accumulated correlation is enough. The speed-up factor is L, which is the message size.

4 Experimental Results

We tested the proposed embedding algorithm for several standard test images. Message sequences are randomly generated for each experiment. The message length, L, is either 1024 for 256 × 256 images, and is 4096 for 512 × 512 images. The orthogonal arcs are the column vectors of a 16 × 16 Hadamard matrix. The values of α and β used in the experiment are both 0.01.

Evaluation of the performance is performed with respect to two criteria. The first is the fidelity. For this, we measured the Watson distance [10] in the watermarked image. The second criterion is the robustness against attacks. For this, we measured the bit error rate (BER).

4.1 Fidelity Test

In Table 1, Watson distance (with R_0=3.5) is compared for several informed watermarking methods [3][7][11] under similar operating conditions. The proposed method provides lower Watson distance.

Table 1. Comparison of Watson distance

	Proposed	Miller	Lin	Ward
Average	49	86	55	50

4.2 Robustness Test

With the proposed informed embedding, BER is zero in the absence of intentional attacks. BER stays at zero until the strength of attack reaches a certain level. We examined the BER performance of our method under various types of attacks. The tested attacks are: additive Gaussian noise, lowpass filtering, valumetric scaling and lossy compression.

Additive White Gaussian Noise
BER under additive Gaussian noise is shown in Figure 5. With the proposed embedding, BER stays at zero until the standard deviation of the Gaussian noise, σ , reaches 5. Even beyond $\sigma = 5$, BER stays at less than 1% while BER for other methods is as high as 4%. In summary, the proposed embedding provides superior robustness against Gaussian noise to other methods.

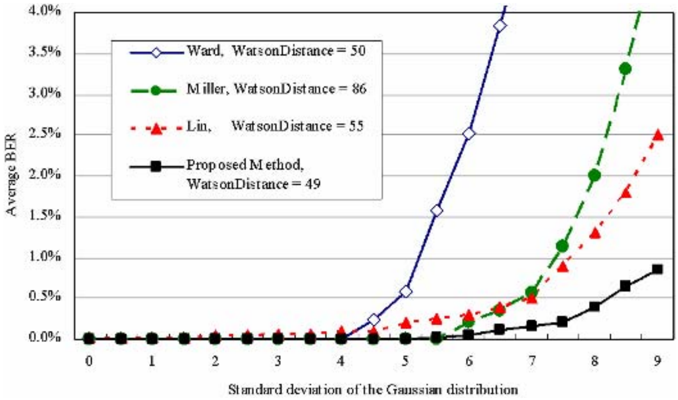


Fig. 5. BER under gaussian noise attack

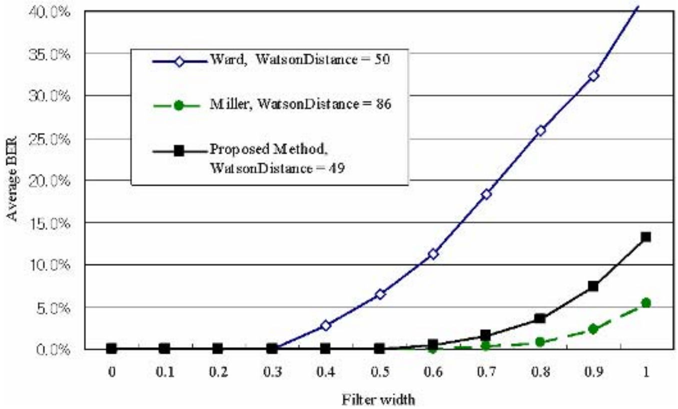


Fig. 6. BER under lowpass filtering

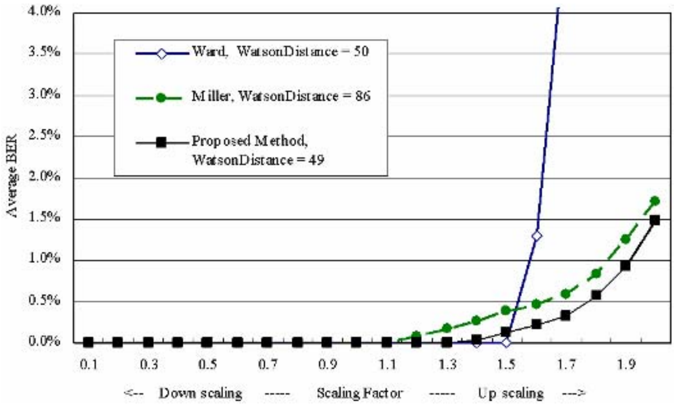


Fig. 7. BER under valuemetric scaling

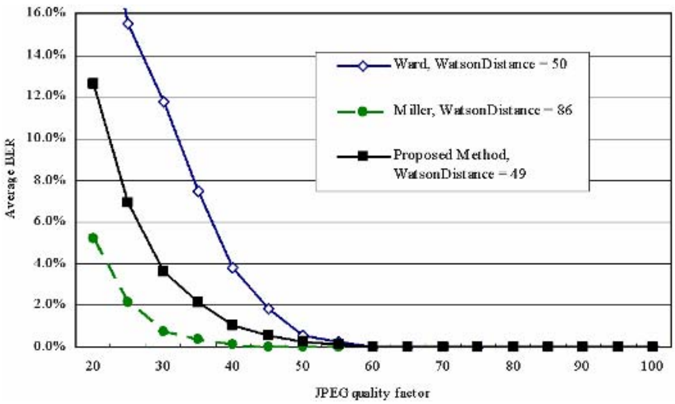


Fig. 8. BER under JPEG compression

Lowpass Filtering
BER under lowpass filtering is shown in Figure 6. The lowpass filter has a size of 21 × 21. BER is measured for the filter width ranging from $\sigma = 0.1 \sim 1.0$. Performance of the proposed method is between [3] and [11]. The proposed method has inferior performance to [3] especially at heavier filtering because it uses larger (16) number of DCT coefficients than in [3] (12).

Valumetric Scaling
BER under valumetric scaling is shown in Figure 7. The scale factor, ν, is ranging from $0.1 \sim 2.0$. When ν is less than 1.0 (downscaling), BER is virtually zero. When ν is larger than 1.0 (upscaling), the proposed embedding provides lower BER, compared to others.

Lossy Compression
BER under JPEG compression is shown in Figure 8. Lower qualify factor (QF) means heavier JPEG compression. BER performance is examined for QF from 20∼100. The proposed embedding provides BER $\leq 1\%$ for QF=40. Inferior performance is due to the larger number of DCT coefficients, as in LPF filtering.

4.3 Speedup Test

Miller's trellis-code embedding is very slow due to the repeated Viterbi decoding at each iteration of $\mathbf{v}$. The speedup factor by the proposed embedding is L. In experiments, the average time for embedding $L(= 1024)$ bits into a 256 × 256 image is 1200 secs [3], 2 secs [7] and 1 sec (proposed method).

5 Conclusions

We present a dirty-paper trellis-code watermarking algorithm with orthogonal arcs. Though the underlying trellis structure is similar to Millers', the orthogonal arcs provide two advantages over the original method. The first one is a speedup with a factor of L. The original trellis-code with random-valued arcs is very slow since modification of the cover work is performed on path-level. The proposed deterministic algorithm with orthogonal arcs is much faster since just incremental arc-level updating of $\mathbf{u}$ is enough for the embedding of each bit.

Experimental results also show that the proposed method provides higher fidelity and lower BER against various types of attacks, compared with other conventional informed watermarking methods.

Acknowledgment. The authors wish to thank the Brain Korea 21 Project of MOE and the Next-generation Digital Convergence Platform Research Fund 2005 of MOCIE for supporting this study.

References

1. Costa, M., "Writing on Dirty Paper," IEEE Trans. Information Theory, vol.29, pp. 439-441, May, 1983
2. Chen, B and Wornell, G., "Digital watermarking and Information embedding using dither modulation," in Proc. of 2nd WMSP, IEEE, pp.273-278, 1998
3. Miller, M., Doërr, G., and Cox, I., "Applying Informed Coding and Embedding to Design a Robust High-Capacity Watermark.," IEEE Tran. Image Processing, vol.,13, pp.792-807, June, 2004
4. Cox, I., Miller, M., and Bloom, J., "Digital Watermarking, Morgan Kaufmann," 2001
5. Kuk., H. and Kim., Y., "Performance Improvement of Order Statistical Patchwork," LNCS, vol. 2939, Springer-Verlag, pp.246-262, Oct. 2003
6. Moulin, P. and O'Sullivan, J., "Information-Theoretic Analysis of Information Hiding," in Proc. of ISIT, IEEE, Jun., 2000, Italy
7. Lin, L., Doërr, G., Cox, I., and Miller, M., "An Efficient Algorithm for Informed Embedding of Dirty-Paper Trellis Codes for Watermarking," in Proc. of ICIP, IEEE, Sep., 2005, Italy
8. Abrardo, A. and Barni, M., "Informed Watermarking by Means of Orthogonal and Quasi-Orthogonal Dirty Paper Coding," IEEE Tran. Signal Processing, vol., 53, IEEE, pp. 824-833, Feb., 2005
9. Abrardo, A. and Barni, M., "Fixed-Distortion Orthogonal Dirty Paper Coding for Perceptual Image Watermarking," LNCS vol. 3200, pp. 52-66, Springer-Verlag, May, 2004
10. Watson, A., "DCT Quantization Matrices Optimized for Individual Images," Human Vision, Visual Processing, and Digital Display IV, vol. SPIE-1913, pp. 202-216, 1993.
11. Coria-Mendoza , L., Nasiopoulos, P., Ward, R., "A Robust Watermarking Scheme Based on Informed Coding and Informed Embedding," in Proc. of ICIP, IEEE, Sep., 2005, Italy

An Efficient Object Tracking Algorithm with Adaptive Prediction of Initial Searching Point*

Jiyan Pan, Bo Hu, and Jian Qiu Zhang

Dept. of E. E., Fudan University. 220 Handan Road, Shanghai 200433, P.R. China
{jiyanpan, bohu, jqzhang01}@fudan.edu.cn

Abstract. In object tracking, complex background frequently forms local maxima that tend to distract tracking algorithms from the real target. In order to reduce such risks, we utilize an adaptive Kalman filter to predict the initial searching point in the space of coordinate transform parameters so that both tracking reliability and computational simplicity is significantly improved. Our method tracks the changing rate of the transform parameters and makes prediction on future values of the transform parameters to determine the initial searching point. More importantly, noises in the Kalman filter are effectively estimated in our approach without any artificial assumption, which makes our method able to adapt to various target motions and searching step sizes without any manual intervention. Simulation results demonstrate the effectiveness of our algorithm.

Keywords: Object tracking, coordinate transform, initial searching point, adaptive Kalman filter.

1 Introduction

Object tracking has been widely applied to video retrieval, robotics control, traffic surveillance and homing technologies. A lot of object tracking algorithms have been reported in literatures, and among them the template matching algorithms has drawn much attention [1]-[6]. In such algorithms, target is modeled by a template, and is tracked in a video sequence by matching candidate image regions with the template through coordinate transforms. The set of transform parameters that yield the highest similarity between the template and the mapped image region of the current frame represents the geometric information of the target.

The performance of object tracking heavily depends on whether the search for the optimal transform parameters can be executed effectively. Many fast searching algorithms have been proposed in an effort to increase the accuracy of searching results

* This work is part of Research on Key Scientific and Technologic Problems of Molecular Imaging and is supported by National Basic Research Program 973 under Grant 2006CB705700.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1113 – 1122, 2006.

while reducing computational complexity. Typical algorithms include Three Step Search (TSS) [7], 2D-Log Search (2DLS) [8], Block-based Gradient Descent Search (BBGDS) [9], and Lucas-Kanade algorithm [1].

For all the algorithms mentioned above, the distraction of local minima is always a serious problem frequently leading to the failure to find the real coordinate transform parameters. Ideally, the image region where real target occupies in the current frame should render the largest similarity measure and therefore unambiguously make itself stand out against the other parts of the frame. When background is cluttered, however, some nearby objects also generate comparable similarity measure and hence confuse tracking algorithms. When searching for optimal coordinate transform parameters, tracking algorithms frequently find themselves trapped into local maxima produced by background objects and other interferences.

Such a situation can be improved by predicting the initial searching point in the space of transform parameters for the next frame and reducing searching range to ensure unimodalilty of the similarity measure. Since most local maxima in the transform parameter space reside some distance from the global maximum where the target locates, the risk of being trapped into local maxima can be substantially reduced if the initial searching point is in the close vicinity of the global maximum. This requires a good prediction of the geometric status of the target in each frame.

In the realm of object tracking, Kalman filters have been used in literatures [6], [11], [12], but few of them serve the purpose of predicting the initial searching point and enhancing tracking performance for the next frame. Besides, the model noises are fixed and determined empirically. In this paper, we propose an approach which employs Kalman filter to track the changing rate of the transform parameters instead of directly filtering their values. Then we select the predicted parameters as the initial searching point for the next frame. More importantly, after analyzing the cause of the model noises in the Kalman filter, we propose an effective method to estimate the power of those noises. As a result, the Kalman filter in our approach can automatically adapt to various target motions and searching step sizes. Experimental results indicate that the proposed method can achieve extremely high accuracy of predicting parameters and hence a significant decrease in the risk of being distracted by background interferences, as well as a considerable drop in computational burden.

The remainder of this paper is organized as follows. Section II focuses on the adaptive Kalman prediction of the initial searching point in the transform parameter space after a brief review of object tracking algorithms based on template matching. Experimental results are included in Section III. The paper is concluded in Section IV.

2 Adaptive Prediction of the Initial Searching Point

2.1 Object Tracking Based on Template Matching

The object (or target) to be tracked is characterized by an image called template which is generally extracted from the first frame of a video sequence. In subsequent frames of the video sequence, the template is mapped to the coordinate system of the frames by coordinate transforms. A searching algorithm tries various combinations of transform parameters to find a set of transform parameters that maximize the similarity between the template and the mapped region of the current frame:

$$\mathbf{a}_m = \underset{\mathbf{a}}{\arg\max} \quad \mathrm{sim}\{I[\varphi(\mathbf{x};\mathbf{a})], T(\mathbf{x})\} \tag{1}$$

where $T(\boldsymbol{x})$ is the grey scale value of a template pixel located at $\boldsymbol{x}$ in the template coordinate system, $I(\boldsymbol{y})$ is the grey scale value of a frame pixel located at $\boldsymbol{y}$ in the frame coordinate system, $\varphi(\boldsymbol{x};\boldsymbol{a})$ is the coordinate transform with parameter vector $\boldsymbol{a}$, sim$\{I,T\}$ is a function that measures the degree of similarity between images I and T. Typical examples of sim$\{I,T\}$ include the normalized linear correlation or the inverse of SSD (sum of squared difference) between I and T [13]. $\boldsymbol{a}_m$ is the transform parameter vector that the searching algorithm assumes to be the one corresponding to correct geometric information of the target.

The type of the coordinate transform is determined by its parameter vector $\boldsymbol{a}$. For the coordinate transform that consists of translation, scaling and rotation, $\boldsymbol{a}$ has four components and $\varphi(\boldsymbol{x};\boldsymbol{a})$ can be written as

$$\varphi(\boldsymbol{x};\boldsymbol{a}) = a_1 \begin{bmatrix} \cos a_2 & \sin a_2 \\ -\sin a_2 & \cos a_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_3 \\ a_4 \end{bmatrix} . \tag{2}$$

Generally speaking, $\varphi(\boldsymbol{x};\boldsymbol{a})$ can have arbitrarily large number of parameters and hence describe extremely complex object motions. Yet the model described by (2) is sufficient for most real-world tracking applications.

2.2 Predicting the Initial Searching Point

In order to predict the initial searching point in the transform parameter space, possible value of each transform parameter in the next frame has to be predicted. Since the frame rate is relatively high, we can reasonably assume the changing rate of each parameter does not alter abruptly over adjacent frame intervals. What brings uncertainty to the changing rate is the influence of arbitrary motion of the target. Such an influence brings about fluctuation of the changing rate of the transform parameters, and thus can be regarded as noise. We employ an adaptive Kalman filter to track the changing rate of the parameters. Such a method is especially instrumental in predicting, not just smoothing, the geometric status of the target. Since different transform parameters describe independent aspects of target motion, they can be predicted separately. The discussion below therefore focuses on one parameter alone and it can be applied to the other parameters trivially.

The state transition equation and the measurement equation for the changing rate of a coordinate transform parameter a are

$$v(n) = v(n-1) + u(n-1) \ , \tag{3}$$

$$v_m(n) = v(n) + w(n) \ , \tag{4}$$

where $v(n)$ is the changing rate of the parameter defined as $a(n)$-$a(n$-1), $v_m(n)$ is the measured changing rate of the parameter, which is actually the increment of the result of parameter search in (1), $u(n)$ is the cause of the fluctuation of $v(n)$ and is white with the power of $\sigma_u^2(n)$, and $w(n)$ is the measurement noise resulting from the limit in the precision of the searching step size for the parameter a. It is also white, with the power of $\sigma_w^2(n)$.

Suppose $\hat{v}_P(n)$ is the prediction of v after the measurement up to frame n-1 is available, and $\hat{v}_E(n)$ is the estimate of v after the measurement up to frame n is acquired. If $e_P(n)$ denotes the prediction error of v and $e_E(n)$ represents the estimation error of v, the following equations hold:

$$v(n-1)=\hat{v}_E(n-1)+e_E(n-1) \ , \tag{5}$$

$$v(n)=\hat{v}_P(n)+e_P(n) \ . \tag{6}$$

Since the state transition coefficient in (3) is one, the estimate of v at frame n-1 serves as the prediction of v at frame n:

$$\hat{v}_P(n)=\hat{v}_E(n-1) \ . \tag{7}$$

From (3), and (5) to (7), the relationship between the prediction and the estimation errors can be derived:

$$e_P(n)=e_E(n-1)+u(n-1) \ . \tag{8}$$

As $e_E(n$-1) is uncorrelated with $u(n$-1), the additive relationship remains for the power of the signals in (8):

$$\sigma_P^2(n)=\sigma_E^2(n-1)+\sigma_u^2(n-1) \tag{9}$$

Where σ_P^2 and σ_E^2 are the power of prediction error and estimation error, respectively.

According to the theory of Kalman filtering [10], the optimal Kalman gain can be expressed as

$$G(n)=\frac{1}{1+\sigma_w^2(n)/\sigma_P^2(n)} \tag{10}$$

where the increase in the prediction error or the decrease in the measurement noise will lead to the rise in the Kalman gain.

After the measured value of v is obtained at frame n, the estimated value of it can be calculated using its predicted value and the Kalman-gain-weighted innovation:

$$\begin{aligned}\hat{v}_E(n)&=\hat{v}_P(n)+G(n)[v_m(n)-\hat{v}_P(n)]\\&=\hat{v}_P(n)+G(n)\alpha(n)\end{aligned} \tag{11}$$

where $\alpha(n)=v_m(n)-\hat{v}_P(n)$ is the innovation at frame n.

Updating the estimate of v leads to the renewal of estimation error as

$$\sigma_E^2(n)=[1-G(n)]\sigma_P^2(n) \cdot \tag{12}$$

(7) and (9) to (12) form a complete iteration to update the prediction of v.

After the predicted value of v for frame n+1 is obtained by applying (7) after (11), the prediction of a at frame n+1 can be written as

$$\hat{a}_P(n+1) = a_m(n) + \hat{v}_P(n+1) \tag{13}$$

where $\hat{a}_P(n+1)$ is the prediction of a at frame n+1, and $a_m(n)$ is the searching result of a at frame n. $\hat{a}_P(n+1)$ is usually very close to the real value of $a(n+1)$ and the initial searching point for a is therefore selected as $\hat{a}_P(n+1)$.

2.3 Estimating the Power of the Model Noises

Although the equations listed above seem to have solved our problem, the power of the two model noises, $\sigma_u^2(n)$ and $\sigma_w^2(n)$, remain to be estimated. Correct evaluation of them plays a key role in obtaining a proper Kalman gain and thus directly determines the performance of the Kalman filter. In the remainder of this section we would like to describe our approach to estimate $\sigma_u^2(n)$ and $\sigma_w^2(n)$.

As is mentioned before, the measurement noise is caused by the non-infinitesimal searching step size in looking for the optimal coordinate transform parameters. For simplicity of notation, we denote $a(n)$ as a_n. Suppose the step size for searching the parameter a_n is Δ, and the searching result is $a_{m,n}$. It is reasonable to assume that the true value of a_n is uniformly distributed over an interval of Δ centered at $a_{m,n}$; that is, the density of the true value of a_n is

$$p_n(a_n) = \begin{cases} 1/\Delta, & |a_n - a_{m,n}| \le \Delta/2 \\ 0, & elsewhere \end{cases}. \tag{14}$$

The power of searching error of a_n can be expressed as follows:

$$\begin{aligned} \sigma_a^2 &= \mathrm{E}\{(a_{m,n} - a_n)^2\} \\ &= \int_{-\infty}^{\infty} (a_{m,n} - a_n)^2 p_n(a_n) da_n \\ &= \int_{a_{m,n}-\Delta/2}^{a_{m,n}+\Delta/2} (a_{m,n} - a_n)^2 \frac{1}{\Delta} da_n \\ &= \frac{\Delta^2}{12} \end{aligned}. \tag{15}$$

Since $v(n)$ is the changing rate of a_n, it is evident that

$$v(n) = a_n - a_{n-1}, \tag{16}$$

$$v_m(n) = a_{m,n} - a_{m,n-1}. \tag{17}$$

Taking (4), (16) and (17) into consideration, we can derive the power of measurement noise $\sigma_w^2(n)$ as follows:

$$\begin{aligned} \sigma_w^2 &= \mathrm{E}\{w^2(n)\} = \mathrm{E}\{(v_m(n) - v(n))^2\} \\ &= \mathrm{E}\{(a_{m,n} - a_n)^2\} + \mathrm{E}\{(a_{m,n-1} - a_{n-1})^2\} \\ &\quad - 2\mathrm{E}\{(a_{m,n} - a_n)(a_{m,n-1} - a_{n-1})\} \end{aligned}. \tag{18}$$

As the parameter searching processes at different frames are uncorrelated, the cross term of (18) is zero. Considering (15), we can reduce (18) to

$$\sigma_w^2 = \mathrm{E}\left\{\left(a_{m,n} - a_n\right)^2\right\} + \mathrm{E}\left\{\left(a_{m,n-1} - a_{n-1}\right)^2\right\} = \frac{\Delta^2}{12} + \frac{\Delta^2}{12} = \frac{\Delta^2}{6} \ . \tag{19}$$

From (19) we can infer that having a finer searching step size can reduce the power of the measurement noise, which is just as expected.

The estimation of $\sigma_u^{\ 2}(n)$, however, is not as straightforward since the motion of the target can be arbitrary. Yet we can still acquire its approximate value by evaluating the power of the innovation $\alpha(n)$. Considering (3), (4), (5) and (7) simultaneously, one can immediately get the following equation which relates the innovation with the estimation error and the two model noises:

$$\alpha(n) = e_E(n-1) + u(n-1) + w(n) \ . \tag{20}$$

The uncorrelatedness among the right-hand terms in (20) yields

$$\sigma_\alpha^2(n) = \sigma_E^2(n-1) + \sigma_u^2(n-1) + \sigma_w^2(n) \tag{21}$$

where $\sigma_\alpha^{\ 2}(n)$ is the power of the innovation and can be approximated as

$$\sigma_\alpha^2(n) = \frac{1}{N} \sum_{k=n-N+1}^{n} \left[v_m(k) - \hat{v}_P(k)\right]^2 \ . \tag{22}$$

N is the number of frames over which the power of the innovation is averaged to obtain its approximate expectation.

Combining (19), (21) and (22), we can acquire the estimation of $\sigma_u^{\ 2}(n)$ as follows:

$$\sigma_u^2(n) = \frac{1}{N} \sum_{k=n-N+2}^{n+1} \left[v_m(k) - \hat{v}_P(k)\right]^2 - \sigma_E^2(n) - \frac{\Delta^2}{6} \tag{23}$$

Where $\sigma_E^{\ 2}(n)$ is calculated online in the iterations of Kalman filtering.

So far we have derived the expressions for estimating the power of the two model noises. By doing so, we do not have to assign any empirical values to those noises as most conventional approaches do. As a result, the method we have proposed can be applied to video sequences with various characteristics of target motion and searching algorithms with different searching step sizes, without any need to tune the Kalman filter manually.

The last step remaining is to initialize the filter. Since we have no information regarding target motion at the very beginning, it is natural to set the initial values of both $\hat{v}_E$ and σ_E^2 to be zero:

$$\hat{v}_E(0) = 0 \ , \ \sigma_E^2(0) = 0 \ . \tag{24}$$

3 Experimental Results

In order to examine how the adaptive prediction of the initial searching point in the transform parameter space can improve the performance of object tracking, we compare the tracking results of two algorithms that are exactly the same in every other aspect except that the first algorithm selects the transform parameters predicted by our proposed method as the initial searching point for the next frame, and the other algorithm just takes the parameters found in the current frame as the initial searching point for the next frame. For simplicity, we denote the algorithm with adaptive prediction of initial searching point as Algorithm 1, and the other one is represented by Algorithm 2. The model of object motion includes translation and scaling. In both algorithms, the searching step size is 1 pixel for horizontal location and vertical location, and 0.05 for scale. Both algorithms select the inverse of SSD as the similarity function [2], and use gradient descent search algorithm to look for optimal transform parameters. Adaptive Kalman appearance filter is employed to update the template.

Figs. 1 to 3 illustrate how well our proposed method predicts the coordinate transform parameters in the next frame. We apply Algorithm 1 to a video sequence where the target undergoes much motion both in spatial locations and scales. Both actual and predicted values of the coordinate transform parameters for every frame are plotted in the same figure.

It can be seen from the figures that our method gives a very precise prediction of what the parameters are going to be in the next frame. The average distance between the initial searching point and the actual point in transform parameter space reduces from 2.7398 to 0.9632 when we use Algorithm 1 instead of Algorithm 2. Such a significant drop in the searching distance is extremely beneficial to tracking algorithms in terms of enhancing tracking stability and decreasing computational burden, as will be demonstrated in the following experimental results.

Fig. 4 and Fig. 5 exemplify considerable improvement of tracking stability when using the adaptive prediction of the initial searching point. When the initial searching point is much closer to the actual point in transform parameter space, tracking algorithms are less likely to be distracted by local maxima resulting from cluttered background, similar objects, or other interferences. This fact is confirmed by our experiments in which we deliberately choose a video sequence that has a vehicle running on a dark road at night. Due to the darkness, the vehicle is blurred and is somewhat similar to the road. When we apply Algorithm 2 to track the vehicle, it is not long before the algorithm loses the target because of being distracted by interferences from the road, as is shown in Fig. 4. Algorithm 1, however, successfully locks on the target throughout the sequence as is demonstrated in Fig. 5. The region in the lower right corner of each frame is the overlapped template.

Computational burden can also be greatly saved by the adaptive prediction of the initial searching point. Since the distance between the initial searching point and the final result point is substantially reduced, it takes searching algorithms in (1) much fewer trials to reach a final status, and computational complexity is therefore considerably reduced. Fig. 6 shows the parameter searching trial times of both algorithms. The right chart of Fig. 6 demonstrates the case where target has relatively high motion. The saving of computational burden is as high as 66.8%. Even in the case where target has low motion that is illustrated in the left chart of Fig. 6, Algorithm 1 can still lower computational complexity by 27.1%.

Since only scalar calculations are involved in the adaptive Kalman prediction of the initial searching point, the proposed algorithm can be implemented real time at a rate of 30fps using C codes on a Pentium-4 1.7GHz PC.

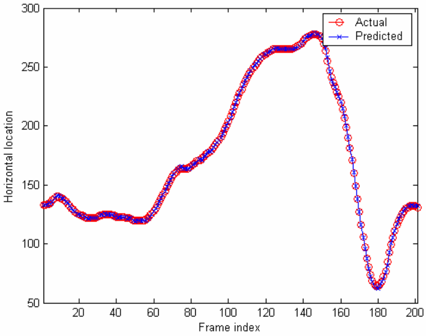


Fig. 1. Curves of the horizontal location of the target. The curve with circles represents the actual horizontal target location of every frame, and the curve with crosses depicts the predicted horizontal target location before every new frame is input.

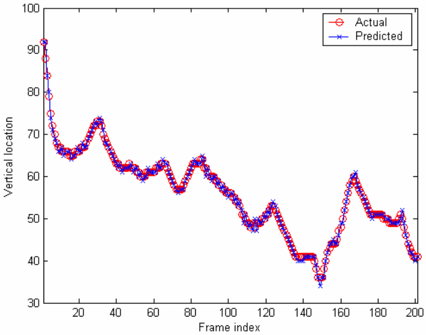


Fig. 2. Curves of the vertical location of the target. The meanings of different types of curves are the same as in Fig. 1.

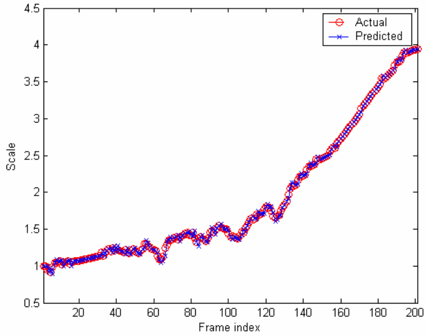


Fig. 3. Curves of the target scale. The meanings of different types of curves are the same as in Fig. 1.

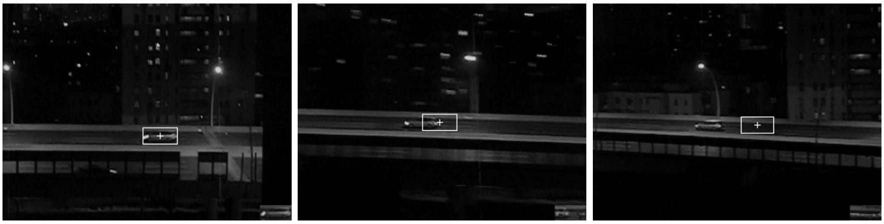


Fig. 4. Algorithm 2 fails to keep track of the vehicle when facing strong interferences from the background. Frame 1, frame 23 and frame 50 are displayed from left to right.

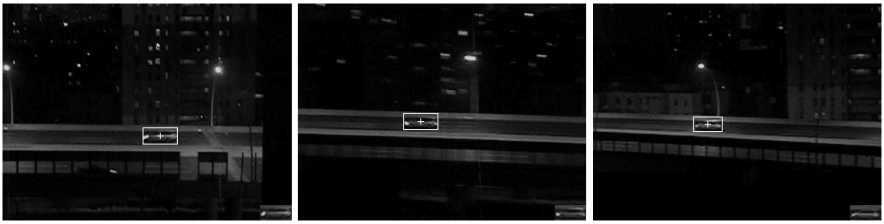


Fig. 5. Algorithm 1 tracks the vehicle perfectly all the time in spite of the existence of strong interferences from the background. Frame 1, frame 23 and frame 50 are displayed from left to right.

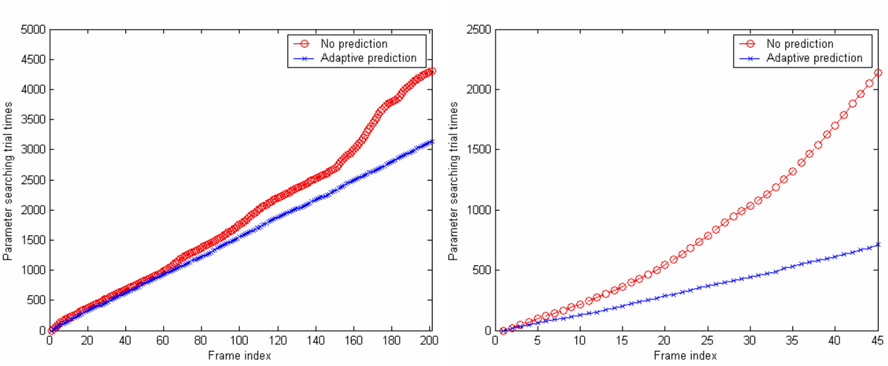


Fig. 6. Curves of parameter searching trial times over frame indices. The curves with circles show the result of Algorithm 2, and the curves with crosses illustrate the result of Algorithm 1. The left chart demonstrates the case where target has low motion, and the right chart, high motion.

4 Conclusion

In this paper we propose an algorithm which adaptively predicts possible coordinate transform parameters for the next frame and selects them as the initial searching point when looking for the real transform parameters. By doing so, tracking algorithms have less risk of being distracted by local maxima resulting from interferences, and tracking performance is thus improved. We use an adaptive Kalman filter to achieve this purpose, but instead of directly filtering the values of transform parameters, we apply the Kalman filter on the changing rate of those parameters to effectively predict their future values. Moreover, we quantitatively analyze the cause of the model noises in the Kalman filter and derive their analytical expressions, so that the Kalman filter in our algorithm is automatically and correctly tuned when the characteristics of target motion change over time, or the searching algorithm uses different searching step sizes. Experimental results show that our proposed algorithm considerably promotes tracking stability while substantially decreasing computational complexity.

References

1. Baker S., Matthews I.: Lucas-Kanade 20 Years on: A Unifying Framework. Int'l J. Computer Vision, Vol. 53, No. 3. (2004) 221-255
2. Matthews I., Ishikawa T., Baker S.: The Template Update Problem. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 26, No. 6. (2004) 810-815
3. Kaneko T., Hori Osamu: Template Update Criterion for Template Matching of Image Sequences. Proc. IEEE Int'l Conf. Pattern Recognition, Vol. 2. (2002) 1-5
4. Peacock A.M., Matsunaga S., Renshaw D., Hannah J.: A. Murray: Reference Block Updating When Tracking with Block Matching Algorithm. Electronic Letters, Vol. 36. (2000) 309-310
5. Smith C., Richards C., Brandt S., Papanikolopoulos N.: Visual Tracking for Intelligent Vehicle-highway Systems. IEEE Trans. on Vehicle Technology, (1996.)
6. Papanikolopoulos N., Khosla P., and Kanade T.: Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision. IEEE Trans. on Robotics and Automation, Vol. 9. (1993) 14-35
7. Wang Y., Ostermann J., Zhang Y.Q.: Video Processing and Communications. Prentice Hall (2002) 159-161
8. Jain J. Jain A.: Displacement Measurement and Its Application in Interframe Image Coding. IEEE Trans. on Communications, Vol. 33. (1981) 1799-1808
9. Liu L.K., Feig E.: A block-based gradient descent search algorithm for block motion estimation in video coding. IEEE Trans. on Circuits and Systems for Video Technology, Vol. 6. (1996) 419-422
10. Brown R.G. and Hwang P.Y.C.: Introduction to Random Signals and Applied Kalman Filtering. John Wiley (1992)
11. Blake A., Curwen R., Zisserman A.: A Framework for Spatio-Temporal Control in the Tracking of Visual Contour. Int'l J. Computer Vision, Vol. 11, No. 2. (1993) 127-145
12. Isard M., Blake A.: CONDENSATION – Conditional Density Propagation for Visual Tracking. Int'l J. Computer Vision, Vol.29, No. 1. (1998) 5-28
13. Sezgin M., Birecik S., Demir D., Bucak I.O., Cetin S., Kurugollu F.: A Comparison of Visual Target Tracking Method in Noisy Environments. Proc. IEEE Int'l Conf. IECON, Vol. 2. (1995) 1360-1365

An Automatic Personal TV Scheduler Based on HMM for Intelligent Broadcasting Services

Agus Syawal Yudhistira[1], Munchurl Kim[1], Hieyong Kim[2], and Hankyu Lee[2]

[1] School of Engineering, Information and Communications University (ICU)
119 Munji Street, Yuseong-Gu, Daejeon, 305-714, Republic of Korea
[2] Broadcasting Media Research Division
Electronic and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Republic of Korea
[1]{agus, mkim}@icu.ac.kr, [2]{hykim, lhk}@etri.re.kr

Abstract. In the future television broadcasting a flood of information from various sources will not always be welcomed by everyone. The need of accessing specific information as required is becoming a necessity. We are interested to make the life of television consumer easier by providing an intelligent television set which can adaptively proposed certain shows to the viewer based on the user historical consumed shows. The method proposed is by utilizing Hidden Markov Model (HMM) to model the user preference of kind of genres the viewer will watch based on recorded genres for several weeks time. The result is satisfactory for users which have middle to high consistent preference of television consumption.

Keywords: Hidden Markov Model (HMM), personalization, multimedia contents.

1 Introduction

Television broadcasting services are entering a new era especially with availability of digital TV, internet, etc. which profoundly affect the TV viewers. Flood of TV programs coming from different sources does not always being welcomed or desirable. Availability of many TV programs at the TV viewer's side entails the difficulty of finding their preferred TV programs. The TV viewers must set aside a considerable amount of their time for searching TV programs and tailoring into their personalized schedules the available published TV program schedules. Worst is the situation when no TV program schedule is available so the TV viewers must find their preferred TV program contents by hopping across the TV channels. The TV viewers can even miss their preferred contents while searching and tailoring TV program schedules for themselves. Our interest is to make it possible the generation of personalized TV program schedules. The personalized TV program scheduler is based on Hidden Markov Model (HMM) by utilizing the TV viewing history data.

The HMM is a well known method to model and predict the processes based on available past behavior information and data. In this paper, our objective is to build a

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1123 – 1132, 2006.

model that predicts the TV watching behaviors in the chronological order of TV genre that a TV viewer is likely to watch. We assume that a TV viewer exhibits a consistent TV watching pattern in his/her TV watching history data. The transition from genre to genre in a day while watching TV can be considered a stochastic process that can be modeled based on the HMM. For the modeling of the chronological order for the watched TV program sequences for each day, we explicitly couple the genre-channel in building our model.

The TV watching history data consists of TV program titles with their respective genres, channels, watched times and durations, etc. To model our automatic personal TV scheduler, the HMM is trained with the TV watching history data for each day. The whole possible TV watching time band is defined from 6:00 P.M. to 12:00 A.M. (midnight) for one day. We assume that the average duration of TV watching time on a specific program is greater than 20 minutes. So the whole time band for one day is segmented into 20 minutes subintervals. The TV watching behavior is observed as follows: (1) it is observed what genres of TV programs a TV viewer has watched every 20 minutes; (2) the transition probabilities of the genres are computed every 20 minutes; (3) the TV watching behavior is then regarded as the most probable transition sequence of genres in TV programs; (4) and an estimate sequence of TV channels can be presented to the TV viewer according to the most probable transition sequence of genres.

This paper is organized as follows: Section 2 briefly introduces the HMM for a self contained purpose; Section 3 presents the modeling of a TV personal scheduler by applying the HMM for the usage history data of TV watching; Section 4 shows the experimental results and we conclude our approach in Section 5.

2 Hidden Markov Model

HMM is a statistical model of sequential data with double stochastic process. One of the processes is not directly observable and termed as hidden. Observation is conducted trough the other stochastic process that produces a sequence of observable data. This observable stochastic process is a probabilistic function of the hidden stochastic process. Although we say the underlying stochastic process is hidden, this does not mean that the states are totally modeled arbitrarily by means of guess. As in our problem, it is possible for us to observe the channel or genre transition directly and to know how many channels and genres are available. But in some other areas, this information may not be available.

One application that extensively uses HMM is speech recognition. But HMM can also be applied for many kinds of problems such as image analysis, language identification, DNA sequencing, handwriting and text recognition, signal processing, climatology and applied also for many other problems. Defined formally, an HMM λ is a 5-tuple of

$$\lambda = (S, V, \Pi, A, B) \tag{1}$$

where $S = \{s_1, s_2, \ldots, s_N\}$ is a finite set of N states, $V = \{v_1, v_2, \ldots, v_M\}$ is a set of M possible symbols that can be emitted from each states, $\Pi = \{\pi_i\}$ are the initial state probabilities, $A = \{a_{ij}\}$ are the state transition probabilities, $B = \{b_j(k)\}$ are the output or emission probabilities (*B* is also called as the confusion matrix). In compact and ordinary form, the definition is written as a triplet:

$$\lambda = (\Pi, A, B) \ . \tag{2}$$

A more detailed of each parameter is as follows:
1. *N*, is the number of states in the model. Thus although the states can be hidden, we are implying that the state is finite.
2. *M*, is the number of distinct observation symbols per state.
3. π_i is the probability that the system starts at state *i* at the beginning.
4. a_{ij} is the probability of going to state *j* from state *i*.
5. lastly, $b_j(k)$ is the probability of emitting symbol v_k at state *j*.
Of course following constraint must be fulfilled as it is in every Markov model:

$$\sum_{i=1}^{N} \pi_i = 1 \tag{3}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \text{for} \quad 1 \le i \le N \tag{4}$$

$$\sum_{k=1}^{M} b_j(k) = 1 \quad \text{for} \quad 1 \le j \le N \ . \tag{5}$$

These constraints assume that the model is a stationary process and will not vary in time. In our case, two special non-emitting states are added: a starting state s_{start} and an ending state s_{end} in addition to the other ordinary emitting states. These states do not have output probability distributions associated with them but they have transition probabilities. s_{start} is always the first state of the model from which transition to other states begins. Thus, the transition probabilities of this state are the initial state probabilities Π itself. s_{end} always comes last toward which the transitions from other states converge. No other transition is possible from s_{end} . For more details, readers are recommended to refer to [1][2]. There are 3 problems to solve with HMM:

1. Given the model λ and observation sequences, how to compute the probability of the particular output? This is a problem of evaluation.
2. Given the model λ and observation sequences, how can we find a sequence of hidden states that best explain the observations? This is a problem of decoding.
3. Given the observation sequences, how can we optimize the model parameters? This is a problem of training.

For training the HMM, we can choose channels or genres as our observation sequences. This training problem is solved by the Baum-Welch algorithm, also known as the forward-backward algorithm. This method consists of two recursion computation: the forward recursion and backward recursion. We use the Baum-Welch algorithm to train our HMM based model [4].

3 Modeling a Personal TV Scheduler Using HMM

The availability of TV watching history data are necessary to employ the HMM: that is what genre of the program contents the person watched, when they have been watched, how long it was watched and in which TV channel it was broadcasted.

In order to implement the HMM to estimate the chronological order of the genre sequences during a day, we choose the genres as the underlying states and the channels as the observations. We assume that TV viewers are allowed to access all the available channels and to watch the program contents in all kinds of genres.

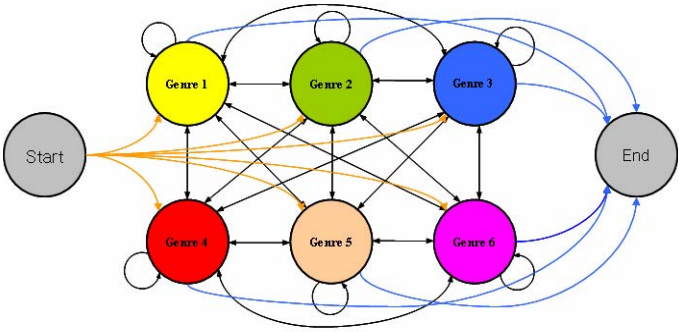


Fig. 1. An example of six states (genre) fully connected Markov model. Both *start* and *end* states are non-emitting, while the other six states emitting observation symbols (channels). The number of states is determined by genre types.

TV viewers usually watches TV program contents on some different channels at anytime. This will present us with an ergodic model as shown in Figure 1. Notice that there is no edge connecting the *start* with *end.*

In order for the HMM to be an effective model, the stochastic process should be stationary, not varying in time. Since TV program schedules do significantly vary in time during the course of the days, weeks and months, a significant range of time can be taken and considered invariant. We divide one day into the time bands of 2 hours, each of which bands is represented by one HMM. Since TV programs are usually broadcast in a weekly cycle, the same time bands in the same days for weeks can be considered invariant so the history data of TV watching from the same day during several weeks can be used for training. This is represented graphically in Fig. 2.

As aforementioned, the time band of two hours is segmented into the sub-time bands of 20 minutes long, thus producing six sub-time bands. Each sub-time band represents an observation instance. In other words, we are checking the possibility of changing TV program contents every 20 minutes. Each 20 minutes period is represented by one genre, which is the one that compose most of the 20 minutes time. For instance, if during that 20 minutes time, the viewer is watching *genre1* for 12 minutes and watching *genre2* for the rest 8 minutes, *genre1* is chosen to represent the period. This will give us a coarse estimation on what genre was actually consumed and what genre is expected. Both the genre and channel information of the day is subject to the representations in 20 minutes unit.

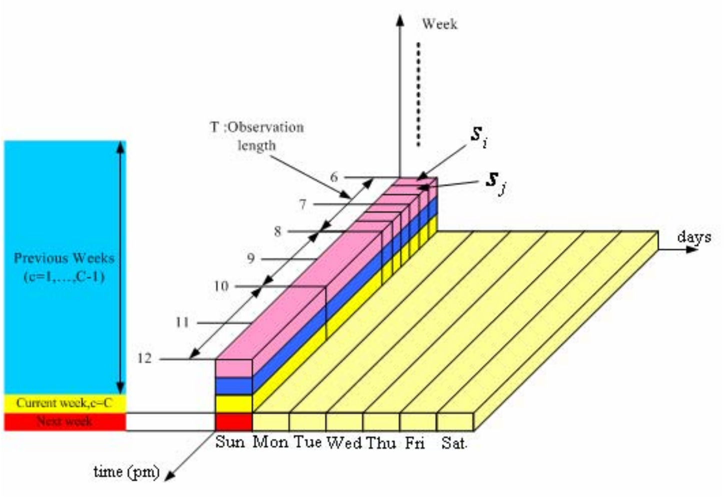


Fig. 2. During a week cycle, a television viewer history is divided into days. For each days the data is divided into 2 hours bands, each represented by an HMM.

For training the HMM, a windowing method is employed. We use training data obtained during 8 weeks for which the TV program schedules have maintained unchanged.

3.1 Initial HMM Parameters

Rabiner states that although in theory the reestimation equations should give values of the HMM parameters which correspond to a local maximum of the likelihood function, experience has shown that either random (subject to the stochastic and the nonzero value constraints) or uniform initial estimates of the Π and $\boldsymbol{A}$ parameters is adequate for giving useful reestimates of these parameters in almost all cases [1].

However for the $\boldsymbol{B}$ parameters, good initial estimates are helpful in the discrete symbol case [1]. Given the sequences of genres $G_r=\{g_r(t)\}$ and channels $C_r=\{c_r(t)\}$, $1\le r \le R$, $1\le t \le T_r$, where here R is the number of weeks inside a window and T_r is total number of genre sub-time band in week r. The component of $\boldsymbol{B}$¸ $b_j(k)$ is

$$b_j(k) = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} match(c_r(t)=s_j, g_r(t)=v_k)}{\sum_{r=1}^{R} T_r} \tag{6}$$

$$\begin{aligned} match(c_r(t)=s_j, g_r(t)=v_k) &= 1 \quad \text{if} \quad c_r(t)=s_j \wedge g_r(t)=v_k \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{7}$$

3.2 Employing the HMM

We employ the HMM for prediction as follows: after the training is completed, a trained HMM is available with new parameters. Following the initial transition Π and parameter $\boldsymbol{A}$, as the case of ordinary Markov process, a sequence of predicted genres will be obtained. From the refined parameter $\boldsymbol{B}$, we can deduct the channel for each corresponding genre in the sequence.

But the genre prediction does not always agree with what is actually broadcasted by each TV channel during the day. Adjustment is made according to the prediction of channels and the actual schedule of each TV channel during the day. For instance, for period 7 P.M. to 7:20 P.M, result of prediction is *genre 1* from *channel A*, but according to the day schedule for *channel A,* it should be *genre 2*, then we correct the prediction to be *genre 2* from *channel A.* We termed this adjustment attempt as synchronization.

4 Experiment and Result

These experiments utilize the data of TV program viewing history from AC Nielson Korea Research Center, recorded by 2,522 people from December 2002 to May 2003. The history data consists of the following database fields:

Table 1. Experiment source database structure

Field Name	Description
id	TV viewer's ID
date	program broadcasting date
dayofweek	a day of the week for program:
subscstart_t	beginning time point of watching a program
subscend_t	ending time point of watching a program
programstart_t	scheduled beginning time of program
programend_t	scheduled ending time of program
channel	channel of program (6 channels)
genre1	genre of program (8 genres)

An attention must be taken regarding the column *genre1*. The column does not provide reliable information on what genres a viewer watches but only provide information of what genres of TV program contents were broadcast before the viewer changes a channel. It is possible though to reconstruct the right schedule table of each channel from the database by aggregating *programstart_t* and *programend_t* of several viewers. We use this resulting table to provide reliable genre information for our experiment performance evaluation.

The six television channels in our experiment include *MBC, KBS1, KBS2, SBS, EBS* and *iTV*. Each of the broadcast channels provides the TV program contents in the eight genres: *Education, drama & movie, news, sports, children, entertainment, information* and *others*. Also in the experiments, we define the whole time band for each day from 6 P.M. to midnight only.

Data for the experiments is taken form several people who exhibit their TV watching behaviors with medium to high consistency week by week. The consistency here means that a TV viewer watches a television almost every week for 6 months, and during the day he/she usually watches the television from 6 PM to midnight. It also means that each of the TV viewers exhibits a similar pattern in the chronological order of TV program genres that he/she has watched during the six months.

Preparation for training as follows: First both channel and genre information during 8 weeks periods is filtered. Those viewing record of less than 5 minutes are regarded as hopping and then excluded. The record then is divided into 2 hours bands and 20 minutes sub-time bands. Thus we can have 6 transitions for each 2 hours time band. Since we have 26 weeks recorded data in our database, we can make experiment to predict 18 consecutive weeks. Each week is predicted by using previous 8 weeks recorded data, and the expected result is validated using actual recorded data of the week we try to estimate as shown in Fig. 3.

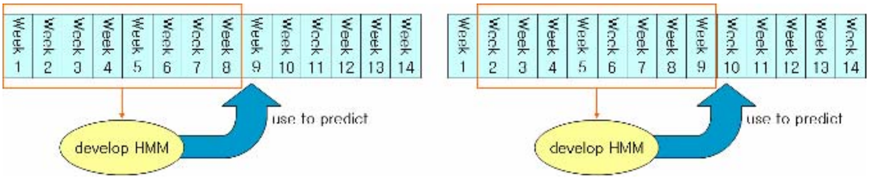


Fig. 3. HMM training scheme using 8 weeks previous recorded history

Using 8 weeks data, parameters of the HMM are built and then trained using channel transitions information. Predicted transitions of channel and genre can be obtained from the trained model.

The evaluation of the trained HMM is shown in Figure 4. The evaluation is made as follows: for each sub-time band of 20 minutes where the prediction match with actual choice the viewer made as recorded in the database, a score is given one. Those that do not match are given zero score. This is done for the sequence of predicted channels and genres, respectively. Consistency of the person is computed with the same method by comparing two consecutive weeks. Given a set of genre transition $G_w = \{g_w(t)\}$, a set of prediction of genre transition $G'_w = \{g'_w(t)\}$, $1 \le w \le W$ and $1 \le t \le T_w$, where here W is the number of experiment weeks we predict and T_w is total number of genre sub-time band in week w. Expected genre transition accuracy is

$$\text{genre prediction accuracy} = \frac{\sum_{w=1}^{W}\sum_{t=1}^{T_w} genre\_match(g'_w(t), g_w(t))}{\sum_{w=1}^{W} T_w} \tag{8}$$

$$\begin{aligned} genre\_match(g'_w(t), g_w(t)) &= 1 \quad \text{if} \quad g'_w(t) = g_w(t) \; . \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{9}$$

The genre consistency is counted as

$$\text{genre consistency} = \frac{\sum_{w=2}^{W}\sum_{t=1}^{T_w} genre\_match(g_{w-1}(t), g_w(t))}{\sum_{w=1}^{W} T_w} \tag{10}$$

$$\begin{aligned} genre\_match(g_{w-1}(t), g_w(t)) &= 1 \quad \text{if} \quad g_{w-1}(t) = g_w(t) \; . \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{11}$$

And given a set of channel transition $C_w = \{c_w(t)\}$, a set of prediction of channel transition $C'_w = \{c'_w(t)\}$, $1 \leq w \leq W$ and $1 \leq t \leq T_w$, where here W is the number of weeks we predict and T_w is total number of channel sub-time band in week w. Expected channel transition accuracy is

$$\text{channel prediction accuracy} = \frac{\sum_{w=1}^{W}\sum_{t=1}^{T_w} channel\_match(c'_w(t), c_w(t))}{\sum_{w=1}^{W} T_w} \tag{12}$$

$$\begin{aligned} channel\_match(c'_w(t), c_w(t)) &= 1 \quad \text{if} \quad c'_w(t) = c_w(t) \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{13}$$

And channel consistency is counted as

$$\text{channel consistency} = \frac{\sum_{w=2}^{W}\sum_{t=1}^{T_w} channel\_match(c_{w-1}(t), c_w(t))}{\sum_{w=1}^{W} T_w} \tag{14}$$

$$\begin{aligned} channel\_match(c_{w-1}(t), c_w(t)) &= 1 \quad \text{if} \quad c_{w-1}(t) = c_w(t) \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{15}$$

Selection of specimen is based on the consistency of the viewer, neglecting the viewer's profile.

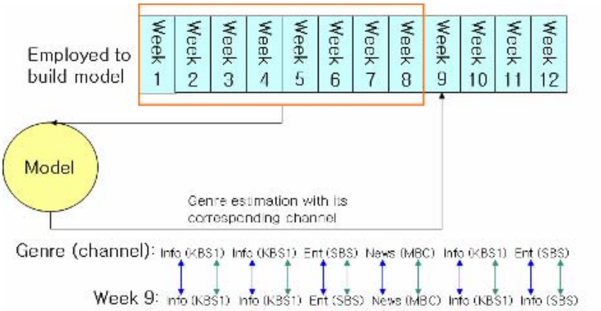

Employed to build model
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Model
Genre estimation with its corresponding channel
Genre (channel): Info (KBS1) Info (KBS1) Ent (SBS) News (MBC) Info (KBS1) Ent (SBS)
Week 9: Info (KBS1) Info (KBS1) Ent (SBS) News (MBC) Info (KBS1) Info (SBS)

Fig. 4. Evaluating expected transition of genres (and channels) of week 9 with actual recorded data of that week

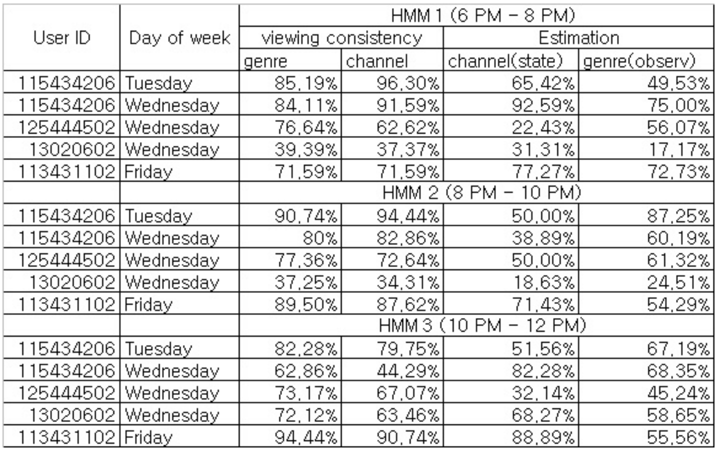
Table 2 and 3 represent the experimental results of specimens under two different scenarios. Table 2 and 3 represent the prediction accuracies with the genre as state and with the channel as state, respectively. Even with the data with high consistency, Table 3 shows lower accuracy results compared to that in Table 2. Especially, the channel prediction is likely to fail along with genre.

Table 2. Prediction accuracy: *Genre* as state and *channel* as observation

User ID	Day of Week	HMM 1 (6 PM – 8 PM)					
		Viewing Consistency		estimation			
		genre	channel	genre	channel	sync genre	
						genre	channel
115434206	Tuesday	85.19%	96.30%	67.59%	98.15%	98.15%	98.15%
115434206	Wednesday	84.11%	91.59%	47.66%	96.26%	96.26%	96.26%
125444502	Wednesday	76.64%	62.62%	17.76%	57.94%	67.29%	57.94%
13020602	Wednesday	39.39%	37.37%	19.19%	36.36%	44.44%	36.36%
113431102	Friday	71.59%	71.59%	70.45%	80.68%	87.50%	80.68%
		HMM 2 (8 PM – 10 PM)					
115434206	Tuesday	90.74%	94.44%	61.11%	81.48%	82.41%	81.48%
115434206	Wednesday	80.00%	82.86%	89.52%	51.43%	54.29%	51.43%
125444502	Wednesday	77.36%	72.64%	51.89%	64.15%	82.08%	64.15%
13020602	Wednesday	37.25%	34.31%	18.63%	36.27%	42.16%	36.27%
113431102	Friday	89.50%	87.62%	35.24%	94.29%	94.29%	94.29%
		HMM 3 (11 PM – 12 PM)					
115434206	Tuesday	82.28%	79.75%	62.03%	86.08%	88.61%	86.08%
115434206	Wednesday	62.86%	44.29%	62.86%	65.71%	75.71%	65.71%
125444502	Wednesday	73.17%	67.07%	32.93%	51.22%	67.07%	51.22%
13020602	Wednesday	72.12%	63.46%	57.69%	70.19%	77.88%	70.19%
113431102	Friday	94.44%	90.74%	47.22%	95.37%	97.22%	95.37%

Under the heading *estimation*, there are three columns. First second column are result of prediction without adjustment. The last column, *sync. genre*, shows accuracy of genre prediction with synchronization as stated in section 3.2. This cannot be done in the scenario which results in Table 3. User ID 115434206 (Wednesday, second model) shows a little anomaly, which can be explained. During 2 hours period the person watched extensively one kind of genre provided from several channels. The high accuracy of genre estimation (without synchronization) shows that the prediction stays in one kind of genre. Yet, for each state, only one channel will be assigned highest probability. The transition of channels is not adequately reflected in this case and thus lowers accuracy result. This can be an indication that 2 hours band is not stationary enough, but overall result is quite good for coarse deduction and content advising. Those that less consistent in TV consumption showed as expected low prediction accuracy. Reason why the second scenario does not provide good result is because the confusion matrix (genre) only reflects distribution of genre broadcasted by the channel. Thus, if the channel failed to produce good prediction the genre will also fail. This is different from the first scenario where the genre probability is summarized from all possible channels available.

Table 3. Prediction accuracy: *Channel* as state and *genre* as observation



User ID	Day of week	HMM 1 (6 PM – 8 PM)			
		viewing consistency		Estimation	
		genre	channel	channel(state)	genre(observ)
115434206	Tuesday	85.19%	96.30%	65.42%	49.53%
115434206	Wednesday	84.11%	91.59%	92.59%	75.00%
125444502	Wednesday	76.64%	62.62%	22.43%	56.07%
13020602	Wednesday	39.39%	37.37%	31.31%	17.17%
113431102	Friday	71.59%	71.59%	77.27%	72.73%
		HMM 2 (8 PM – 10 PM)			
115434206	Tuesday	90.74%	94.44%	50.00%	87.25%
115434206	Wednesday	80%	82.86%	38.89%	60.19%
125444502	Wednesday	77.36%	72.64%	50.00%	61.32%
13020602	Wednesday	37.25%	34.31%	18.63%	24.51%
113431102	Friday	89.50%	87.62%	71.43%	54.29%
		HMM 3 (10 PM – 12 PM)			
115434206	Tuesday	82.28%	79.75%	51.56%	67.19%
115434206	Wednesday	62.86%	44.29%	82.28%	68.35%
125444502	Wednesday	73.17%	67.07%	32.14%	45.24%
13020602	Wednesday	72.12%	63.46%	68.27%	58.65%
113431102	Friday	94.44%	90.74%	88.89%	55.56%

5 Conclusion

In this paper, we propose an automatic personal TV scheduler that is modeled using HMM. The usage history data of TV watching has been used to train and to test the proposed personal TV scheduler. The double stochastic property of HMM is exploited to synchronize the channel preference with the expected genre preference. It provides satisfactory accuracy to predict TV viewer's preference for middle to high consistent TV consumption.

References

1. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition. Proc. IEEE 77 (1989) 257-285.
2. Zhai, Cheng Xiang: A Brief Note on the Hidden Markov Models (HMMs). Department of Computer Science University of Illinois at Urbana-Champaign (2003).
3. Stamp, Mark: A Revealing Introduction to Hidden Markov Models. Department of Computer Science San Jose State University (2004).
4. Young, S., Evermann, G., Gales, M., et al.: The HTK Book. Cambridge University Engineering Department (2005) 6-8, 128-130.

Increasing the Effect of Fingers in Fingerspelling Hand Shapes by Thick Edge Detection and Correlation with Penalization

Oğuz Altun and Songül Albayrak

Yıldız Technical University, Computer Engineering Department, Yıldız, İstanbul, Türkiye
{oguz, songul}@ce.yildiz.edu.tr

Abstract. Fingerspelling is used in sign language to spell out names of people and places for which there is no sign or for which the sign is not known. In this work we describe a method for increasing the effect of fingers in Fingerspelling hand shapes. Hand shape objects are obtained by extraction of representative frames, color segmentation in YCrCb space and angle of least inertia based fast alignment [1]. *Thick edges* of the hand shape objects are extracted with a distance to edge based method. Finally a calculation that penalizes similarity for not-corresponding pixels is employed to correlation based template matching. The experimental Turkish fingerspelling recognition system recognizes all 29 letters of the Turkish alphabet. The train video database is created by three signers, and has a set of 290 videos. The test video database is created by four signers, and has a set of 203 videos. Our methods achieve a success rate of 99%.

Keywords: Turkish Fingerspelling Recognition, Fast Alignment, Angle of orientation, Axis of Least Inertia, Thick Edges, Correlation with Penalization.

1 Introduction

Sign Language is the language used mainly by deaf people and people with hearing difficulties as a visual means of communication by gestures, facial expression, and body language. There are two major types of communication in sign languages: word based and letter based. The first one has word based sign vocabulary, and gestures, facial expression, and body language are used for communicating these words. The second one has letter based vocabulary, and is called fingerspelling. It is used to spell out names of people and places for which there is no sign and can also be used to spell words for signs that the signer does not know the sign for, or to clarify a sign that is not known by the person reading the signer [2].

Sign languages develop specific to their communities and are not universal. For example, ASL (American Sign Language) is totally different from British Sign Language even though both countries speak English [3]. Quite a number of Sign Language recognition systems are reported amongst which are American Sign Language (SL) [4], Australian SL [5], and Chinese SL [6].

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1133 – 1141, 2006.

Previous approaches to word level sign recognition rely heavily on statistical models such as Hidden Markov Models (HMMs). A real-time ASL recognition system developed by Starner and Pentland [4] used colored gloves to track and identify left and right hands. They extracted global features that represent positions, angle of axis of least inertia, and eccentricity of the bounding ellipse of two hands. Using an HMM recognizer with a known grammar, they achieved a 99.2% accuracy at the word level for 99 test sequences. For TSL (Turkish Sign Language) Haberdar and Albayrak [7], developed a TSL recognition system from video using HMMs for trajectories of hands. The system achieved a word accuracy of 95.7% by concentrating only on the global features of the generated signs. The developed system is the first comprehensive study on TSL and recognizes 50 isolated signs. This study is improved with local features and performs person dependent recognition of 172 isolated signs in two stages with an accuracy of 93.31% [8].

For fingerspelling recognition, most successful approaches are based on instrumented gloves, which provide information about finger positions. Lamar and Bhuiyant [9] achieved letter recognition rates ranging from 70% to 93%, using colored gloves and neural networks. More recently, Rebollar et al. [10] used a more sophisticated glove to classify 21 out of 26 letters with 100% accuracy. The worst case, letter 'U', achieved 78% accuracy. Isaacs and Foo [11] developed a two layer feed-forward neural network that recognizes the 24 static letters in the American Sign Language (ASL) alphabet using still input images. ASL fingerspelling recognition system is with 99.9% accuracy with an SNR as low as 2. Feris, Turk and others [12] used a multi-flash camera with flashes strategically positioned to cast shadows along depth discontinuities in the scene, allowing efficient and accurate hand shape extraction. Altun et al. [1] used axis of least inertia base fast alignment and compared different classification algorithms. They achieved 98.83% accuracy out of 174 videos of 29 Turkish alphabet letter. 1NN and SVM were the best classifiers.

In this work, we have developed a signer independent fingerspelling recognition system for Turkish Sign Language (TSL). The representative frames are extracted from sign videos. Hand objects in these frames are segmented out by skin color in YCrCb space. These hand objects are aligned using the angle of orientation based fast alignment method. Then, the aligned object is moved into the center of a minimum bounding square, and resized. Thick edges of the object in minimum bounding square are used for correlation with penalization based template matching.

The remaining of this paper is organized as follows: In Section 2 we describe the representative frame extraction, skin detection by color, our fast alignment method, and thick edge extraction. We explain the correlation with penalization algorithm in Section 3. Section 4 covers the video database we use, and supplies a table of the distribution of number of signers in test and train databases. In Section 5 we evaluate the experimental results, and we supply a table that shows the effects of the methods we use. Section 6 has a discussion of assumptions and future work. Finally, conclusions are addressed in Section 7.

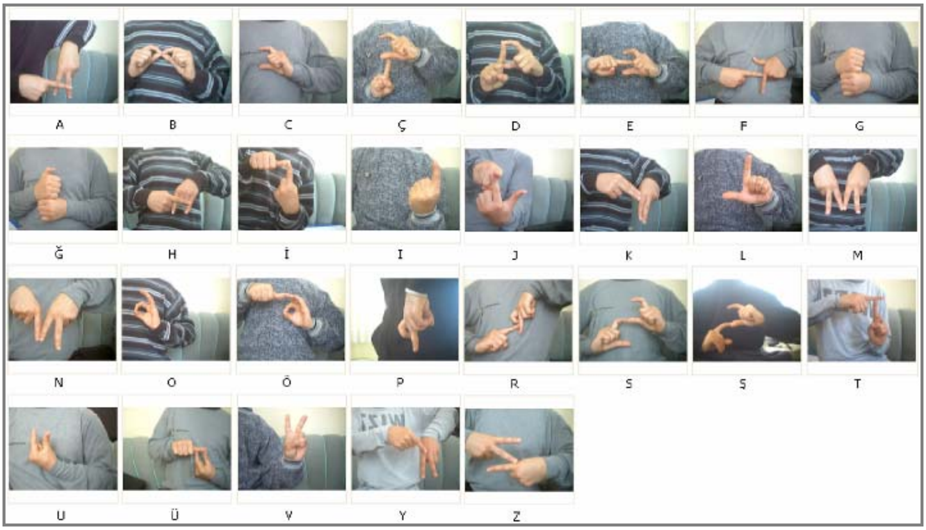


Fig. 1. Representative frames for all 29 letters in Turkish Alphabet

2 Feature Extraction

Contrary to Turkish Sign Language word signs, Turkish fingerspelling signs, because of their static structure, can be discriminated by shape alone by use of a representative frame. To take advantage of this and to increase processing speed, these representative frames are extracted and used for recognition. Fig. 1 shows representative frames for all 29 Turkish Alphabet letters.

In each representative frame, hand regions are determined by skin color. From the binary images that show hand and background pixels, the regions we are interested in are extracted, aligned and resized as explained in [1]. Finally, from those, the thick edges are extracted for template matching.

2.1 Representative Frame Extraction

In a Turkish fingerspelling video, representative frames are the ones with least hand movement. Hence, the frame with minimum distance to its successor is chosen as the representative. Distance between successive frames f and f+1 is given by the sum of the city block distance between corresponding pixels [1]:

$$D^f = \sum_n |\Delta R_n^f| + |\Delta G_n^f| + |\Delta B_n^f|, \qquad (1)$$

where f iterates over frames, n iterates over pixels, R, G, B are the components of the pixel color, $\Delta R_n^f = R_n^{f+1} - R_n^f$, $\Delta G_n^f = G_n^{f+1} - G_n^f$, and $\Delta B_n^f = B_n^{f+1} - B_n^f$.

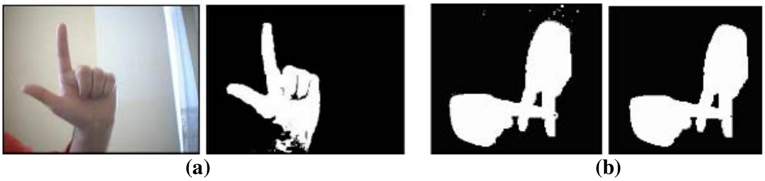


Fig. 2. (a) Original image and detected skin regions after pixel classification, (b) result of the morphological opening

2.2 Skin Detection by Color

For skin detection, YCrCb color-space has been found to be superior to other color spaces such as RGB and HSV [13]. Hence we convert the pixel values of images from RGB color space to YCrCb using (2):

$$Y = 0.299R + 0.587G + 0.114B, Cr = B - Y, Cb = R - Y \cdot \tag{2}$$

In order to decrease noise, all Y, Cr, and Cb components of the image are smoothed with a 2D Gaussian filter given by (3),

$$F(x, y) = (1/2\pi\sigma^2)\exp(-(x^2 + y^2)/2\sigma^2) \tag{3}$$

where σ is the standard deviation.

Chai and Bouzerdom [14] report that pixels that belong to the skin region have similar Cr and Cb values, and give a distribution of the pixel color in Cr-Cb plane. Consequently, we classified a pixel as skin if the Y, Cr, Cb values of it falls inside the ranges $135 < Cr < 180$, $85 < Cb < 135$ and $Y > 80$ (Fig. 2.a).

As in Altun et al. ([1]), after clearing small skin colored regions by morphological opening (Fig. 2.b), skin detection is completed.

2.3 Fast Alignment for Correlation Based Template Matching

Template matching is very sensitive to size and orientation changes. Hence a scheme that can compensate size and orientation changes is needed. Eliminating orientation information totally is not appropriate however, as depicted in Fig. 3. Fig. 3a-b show two 'C' signs that we must be able to match each other, so we must compensate the small orientation difference. In Fig. 3c-d we see two 'U' signs that we need to differentiate from 'C' signs. 'U' signs and 'C' signs are quite similar to each other in shape, luckily orientation is a major differentiator. As a result we need a scheme that

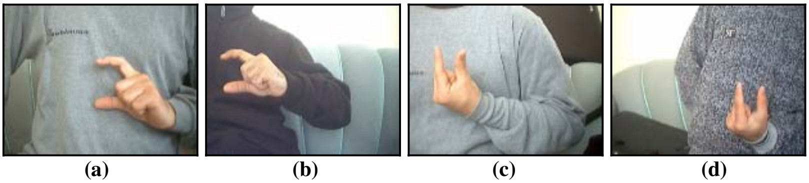


Fig. 3. (a)-(b) The 'C' sign by two different signers. (c)-(d) The 'U' sign by two different signers.

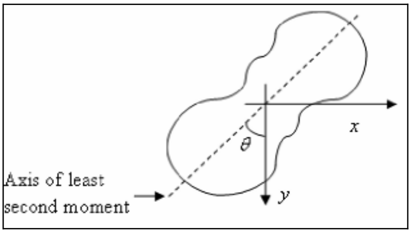


Fig. 4. Axis of least second moment and the angle of orientation *θ*

not only can compensate small orientation differences of hand regions, but also is responsive to large ones.

We use the fast alignment method [1] that makes the *angle of orientation θ* zero. Angle of orientation is the angle between *y* axis and the *axis of least moment* (shown in Fig. 4), and is given by (4):

$$\begin{gathered} 2\theta = \arctan\left(2M_{11}/(M_{20} - M_{02})\right), \\ M_{11} = \sum_x \sum_y xyI(x,y) \\ M_{20} = \sum_x \sum_y x^2 I(x,y) \\ M_{02} = \sum_x \sum_y y^2 I(x,y), \end{gathered} \tag{4}$$

where $I(x,y) = 1$ for pixels on the object, and 0 otherwise.

Bounding square is defined as the smallest square that can completely enclose all the pixels of the object [1]. After putting images in the center of a bounding square, and than resizing the bounding square to a fixed, smaller resolution, the fast alignment process ends (Fig. 5).

2.4 Thick Edge Extraction

Most edge detection algorithms give edges of only one pixel wide. We propose a method to adjust the thickness of the edges without damaging the outer contours of the objects. Our method also adjusts the importance of the edge pixels for correlation.

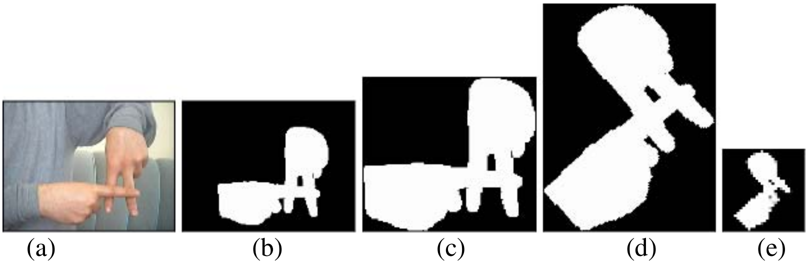


Fig. 5. Stages of fast alignment. (a) Original frame. (b) Detected skin regions. (c) Region of Interest (ROI). (d) ROI rotated according to the angle of orientation. (e) Resized bounding square with the object in the center.

Assume a binary image b where pixels with values equal to zero (zero-pixels) are background, the rest of the pixels (non-zero-pixels) represent objects, and non-zero-pixels have the value *Max* (e.g. 255). Then, given an edge image e produced from binary image by an edge detector, the thick edge image t can be produced by the algorithm in Fig. 6:

- Let x, y, u, and v represent pixel coordinates.
- For each non-zero pixel $b[x,y]$ of the binary image b,
    - Let $e[u,v]$ be the non-zero pixel in edge image whose coordinates u, v has the closest Euclidean distance to the coordinate $[x,y]$. Call that distance $dist[x,y]$.
    - $t[x,y] = Max - steepness*dist[x,y]$,
    - If $t[x,y] < 0$, $t[x,y] = 0$.

Fig. 6. Thick Edge extraction algorithm

The *steepness* helps to adjust the thickness and the relative importance of pixels on edges, as seen in Fig. 7.

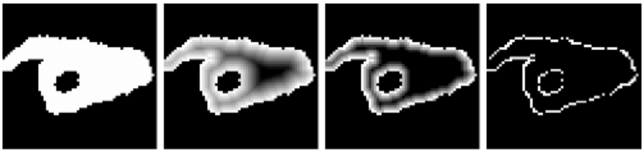


Fig. 7. Thick edges of the same binary image. The *steepness* value increases from left to right.

In a fingerspelling shape, most of the information is in the fingers of a hand, and not in the palm (or back of the hand). Thick edge extraction increases contribution of the fingers to the matching result.

3 Correlation with Penalization

Similarity of two templates is calculated by the algorithm summarized in Fig. 8.

Image pixels take zero or positive values. By using the algorithm in Fig. 8 for correlation based template matching, we not only award the coincidence of two non-zero values, but also penalize the coincidence of a zero with a non-zero value. This ensures that the matching shapes have more matching edges, and less non-matching ones.

```
• For each of the corresponding pixel values first and second of the two templates
    o If one of the first or second is equal to zero, and the other is not,
      $similarity = similarity - (first^2 + second^2)*penalty$
    o Else,
      $similarity = similarity + first * second$
```

Fig. 8. Correlation with penalization algorithm. The factor *penalty* helps in adjusting the amount of penalization.

4 Video Database

The train and test videos are acquired by a Philips PCVC840K CCD webcam. The capture resolution is set to 320x240 with 15 frames per second (fps). While programming is done in C++, the Intel OpenCV library routines are used for video capturing and some of the image processing tasks.

We develop a Turkish Sign Language fingerspelling recognition system for the 29 static letters in Turkish Alphabet. The training set is created using three different signers. For training, they sign a total of 10 times for each letter, which sums up to 290 training videos. For testing, they and an additional signer sign a total of 7 times for each letter, which sums up to 203 test videos. Table 1 gives a summary of the distribution of the train and test video numbers for each signer. Notice that training and test sets are totally separated, and test set has the videos of a signer for whom there is no video in the train set.

Table 1. Distribution of train and test video numbers for each signer

	Signer 1		Signer 2		Signer 3		Signer 4		Total	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
A	4	2	4	2	2	2	0	1	10	7
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Z	4	2	4	2	2	2	0	1	10	7
Total									290	203

5 Experimental Results

Fast alignment, thick edges and correlation with penalization produced a success rate of 99% out of 203 test instances of the 29 Turkish fingerspelling videos. That is, the system gives just two errors.

The effect of thick edge extraction and correlation with penalization can be seen in Table 2. Fast alignment is used in all experiments, and it alone gives only 4 errors. An interesting point is that thick edge extraction decreases the success alone, but increases when combined with correlation with penalization. This shows how useful non-matching edge pixels are for the purpose of distinguishing between different hand shapes.

Table 2. Number of true classifications, number of false classifications and the success rate for different combinations of methods. Fast alignment is applied in all cases.

	Correlation with Penalization			Correlation without Penalization		
	#True	#False	Success Rate (%)	#True	#False	Success Rate (%)
Thick Edges	201	2	99.02	195	8	96.06
No Thick Edges	198	5	97.54	199	4	98.03

6 Discussions and Future Work

Even though we assume the inverse, not all letters in Turkish alphabet are representable by one single frame, 'Ş' being an example. The sign of this letter involves some movement that differentiates it from 'S'. Still, representing the whole sign by one single frame is acceptable since this work is actually a step towards making a full blown Turkish Sign Language recognition system that can also recognize word signs. That system will incorporate not only shape but also the movement, and the research on it is continuing.

The importance of successful segmentation of the skin and background regions can not be overstated. In this work we assume that there is no skin colored background regions and used color based segmentation in YCrCb space. The systems' success depends on that assumption; research on better skin segmentation is invaluable.

Testing and training sets are created by multiple signers, and test set has a signer for whom train set has no video.

The system is fast due to single frame video representation, fast alignment process, and resizing the bounding square to a smaller resolution. The amount of resizing can be arranged for different applications.

Fast alignment, thick edge extraction and correlation with penalization methods are robust to the problem of occlusion of the skin colored regions (e.g. hands) with each other, because we do not try to find individual hands, and because our methods allow us to process occluding hand shapes as one big fingerspelling shape.

The methods presented here would work equally well in the existence of a face in the frame, even though in this study we used a fingerspelling video database ([1]) that has only hand regions in each frame.

Although we demonstrated these methods in the context of hand shape recognition, they are equally applicable to other problems where shape recognition is required, for example to the problem of shape retrieval.

7 Conclusions

A Turkish fingerspelling recognition system is tested and found to have 99.02% success rate.

This high success rate is the result of the fast alignment, thick edge extraction, and correlation with penalization methods. Fast alignment process brings objects with similar orientation into same alignment, while bringing objects with high orientation difference into different alignment. This is a desired result, because for fingerspelling recognition, shapes that belong to different letters can be very similar, and the orientation can be the main differentiator. After the alignment, to resize without breaking the alignment, the object is moved into the center of a minimum bounding square. Thick edge extraction increases the contribution of the thinner parts of the hand shapes (like fingers) to result. This is also desired, since most of the information in a fingerspelling hand shape is in the fingers. As a final step, correlation with penalization helps to account for not only matching pixels, but also pixels that do not match.

Our research on converting this template matching strategy into a local descriptor is continuing.

References

1. Altun, O., Albayrak, S., Ekinci, A., Bükün, B.: Turkish Fingerspelling Recognition System Using Axis of Least Inertia Based Fast Alignment. The 19th Australian Joint Conference on Artificial Intelligence (AI 2006) (2006)
2. http://www.british-sign.co.uk/learnbslsignlanguage/whatisfingerspelling.htm.
3. http://www.deaflibrary.org/asl.html.
4. Starner, T., Weaver, J., Pentland, A.: Real-time American sign language recognition using desk and wearable computer based video. Ieee Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1371-1375
5. Holden, E.J., Lee, G., Owens, R.: Australian sign language recognition. Machine Vision and Applications **16** (2005) 312-320
6. Gao, W., Fang, G.L., Zhao, D.B., Chen, Y.Q.: A Chinese sign language recognition system based on SOFM/SRN/HMM. Pattern Recognition **37** (2004) 2389-2402
7. Haberdar, H., Albayrak, S.: Real Time Isolated Turkish Sign Language Recognition From Video Using Hidden Markov Models With Global Features. Lecture Notes in Computer Science **LNCS 3733** (2005) 677
8. Haberdar, H., Albayrak, S.: Vision Based Real Time Isolated Turkish Sign Language Recognition. International Symposium on Methodologies for Intelligent Systems, Bari, Italy (2006)
9. Lamar, M., Bhuiyant, M.: Hand Alphabet Recognition Using Morphological PCA and Neural Networks. International Joint Conference on Neural Networks, Washington, USA (1999) 2839-2844
10. Rebollar, J., Lindeman, R., Kyriakopoulos, N.: A Multi-Class Pattern Recognition System for Practical Fingerspelling Translation. International Conference on Multimodel Interfaces, Pittsburgh, USA (200)
11. Isaacs, J., Foo, S.: Hand Pose Estimation for American Sign Language Recognition. Thirty-Sixth Southeastern Symposium on, IEEE System Theory (2004) 132-136
12. Feris, R., Turk, M., Raskar, R., Tan, K.: Exploiting Depth Discontinuities for Vision-Based Fingerspelling Recognition. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops(CVPRW'04) (2004)
13. Sazonov, V., Vezhnevetsi, V., Andreeva, A.: A survey on pixel vased skin color detection techniques. Graphicon-2003 (2003) 85-92
14. Chai, D., Bouzerdom, A.: A Bayesian Approach To Skin Colour Classification. TENCON-2000 (2000)

Binary Edge Based Adaptive Motion Correction for Accurate and Robust Digital Image Stabilization

Ohyun Kwon[1,*], Byungdeok Nam[1], and Joonki Paik[2]

[1] DSC R&D Center, Optics & Digital Imaging Business, Samsung Techwin Co., Ltd
145-3, Sangdaewon 1-Dong, Jungwon-Gu, Sungnam-City, Kyungki-Do, Korea, 462-121
kohnann@gmail.com
[2] Image Processing and Intelligent Systems Laboratory, Department of Image Engineering,
Graduate School of Advanced Imaging Science, Multimedia, and Film, Chung-Ang University

Abstract. Digital image stabilization (DIS) becomes one of the most important features in consumer cameras. The image sequence is easily interfered by hand-shaking during acquisition process especially in zoom-in highly. Many of the state-of-the-art algorithms for image stabilization perform well for high patterned and clear images. However, for high noise or low pattern level images, the algorithm performance degrades seriously. In this paper, we propose a novel digital image stabilization algorithm for removal of unwanted motion in high noise or low pattern level images. The proposed algorithm consists of an adaptive Kalman filter for motion prediction and a binary edge transform (BET) based phase correlation (PC) for motion estimation. Since the proposed algorithm is designed for real-time implementation, it can be used as an algorithm for a DIS to preserve a good environment that the photograph has been taken of many commercial applications such as low cost camcorders, digital cameras, CCTV, and surveillance video systems.

Keywords: Digital image stabilization, motion estimation, motion correction, binary edge transform, motion discrimination, Kalman filter.

1 Introduction

A demand for digital image stabilization techniques is rapidly increasing in many visual applications, such as camcorders, digital cameras, and video surveillance systems. Image stabilization refers to a digital image processing technique that registers sequence image, differently out-of-focused by unstable image. Conventional stabilization techniques, such as manual stabilization, through the lens auto-stabilization, and semi-digital auto-stabilization, cannot inherently deal with digital image stabilization. Image stabilization can be realized with fully digital auto-stabilization based on PC estimation and correction.

In this paper, a novel shake adaptive binary edge transformation, which detects most jitter in noisy images and provides superior environment that photograph taken, has been proposed. Furthermore, the proposed BET filter can be implemented with the minimum hardware expense of a single frame memory while the conventional temporal filters require several frame memories to implement. Since stabilization

[*] *Member, IEEE.*

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1142 – 1149, 2006.

algorithms on noisy images often result in noise amplification, the proposed algorithm should be used prior to any image stabilization operation. The proposed algorithm can also be used in any optical instrument to overcome handshake impairments in the acquired image and thereby resulting in better image quality.

The rest of the paper is organized as follows. Existing stabilization techniques and problem formulation are described in section 2. The proposed BET and adaptive motion correction algorithm is described in section 3. Simulation results and comparisons are shown in section 4. Finally, concluding remarks are outlined in section 5.

2 Existing State-of-the-Art Methods

The image stabilization systems can be classified into three major types in [1]: the electronic, the optical [2], and the digital stabilizers. In DIS, various algorithms had been developed to estimate the local motion vectors such as representative point matching (RPM) [3], edge pattern matching (EPM) [4], bit-plane matching (BPM) [5], one bit transform matching (1BTM) [6], hierarchical distributed template matching (HDTM) [7], and others [8][9][10]. The major objective of these algorithms is to reduce the computational complexity, in comparison with full-search block-matching method, without losing too much accuracy and reliability. In general, the RPM and 1BTM can greatly reduce the complexity of computation in comparison with the other methods. However it is sensitive to irregular conditions such as moving objects and intentional panning, etc. Therefore, the reliability evaluation is necessary to screen the undesired motion vectors for the theses method. In [1], an inverse triangle method is proposed to extract reliable motion vectors in plain images which are lack of features or contain large low-contrast area, etc., and a background evaluation model is developed to deal with irregular images which contain large moving objects, etc. Using HDTM method, only useful reference blocks that are indispensable for accurate motion estimation are selected with its reliability and consistency on pose estimation was proposed in [7]. However, these algorithms cannot cover widely various irregular conditions such as the low level images, and it is also hard to determine an optimum partial template block for discrimination in various conditions. Therefore the accurate estimation is necessary in poor conditions.

In the motion correction of DIS, accumulated motion vector estimation [4] and frame position smoothing (FPS) [8] are the two most popular approaches. The accumulated motion vector estimation needs to compromise stabilization and intentional panning preservation since the panning condition causes a steady-state lag in the motion trajectory. The FPS accomplished the smooth reconstruction of an actual long-term camera motion by filtering out jitter components based on the concept of designing the filter with appropriated cut-off frequency or adaptive fuzzy filter to continuously improve stabilization performance. But, these algorithms cannot cover unfeasible motion vector from motion estimation.

3 Proposed Digital Image Stabilization Algorithm

The proposed digital image stabilization algorithm uses the following two procedures to obtain a suitable registered image: (i) Using a binary edge based phase correlation for motion estimation (ii) kalman and motion discrimination (MD) filter for motion correction shown in Fig. 1.

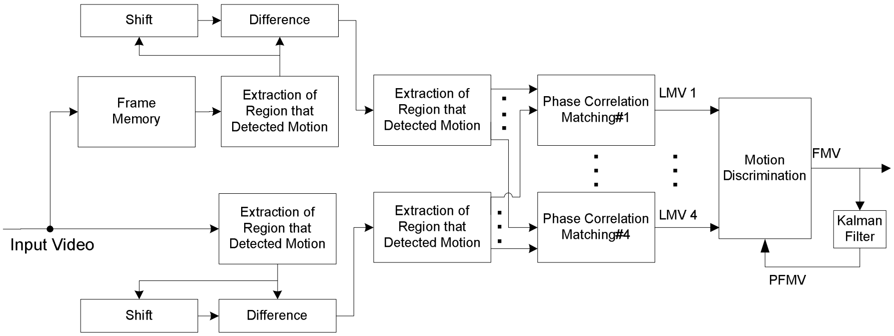

Shift
Difference
Frame Memory
Extraction of Region that Detected Motion
Extraction of Region that Detected Motion
Phase Correlation Matching#1
LMV 1
Motion Discrimination
FMV
Kalman Filter
Input Video
Extraction of Region that Detected Motion
Extraction of Region that Detected Motion
Phase Correlation Matching#4
LMV 4
PFMV
Shift
Difference

Fig. 1. Block diagram of the proposed digital image stabilization system

3.1 Motion Estimation Using Binary Edge Based Phase Correlation

Binary Edge Transformation based phase correlation estimation, in which frames are transformed into binary number/pixel representation by comparing the original image frame against a difference in shifted original image frame version, provides a low complexity and a high accuracy phase correlation motion estimation approach. Simple and fast transfer to edge image frame from original image frame, we propose shift difference edge (SDE) transformation in the form of

$$SDE(i,j) = I(i,j) - I(i-\alpha, j-\alpha). \tag{1}$$

where $SDE(i,j)$ represents the edge image frame using shift difference transformation and α is the constant value for width of edge line. The binary edge image $BE(i,j)$ with some threshold the constructed as

$$BE(i,j) = \begin{cases} 1 & ,SDE(i,j) > th \\ 0 & ,SDE(i,j) \leq th \end{cases} \tag{2}$$

The empirically selected threshold in the range, 0< th <10, provides acceptable results for most unstable images. The block diagram of motion estimation is displayed in Fig.2.

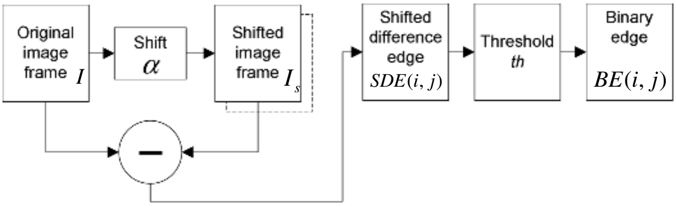


Fig. 2. The motion estimation of the proposed BE transformation

The BE transform result for a sample image frame is displayed in Fig. 3.

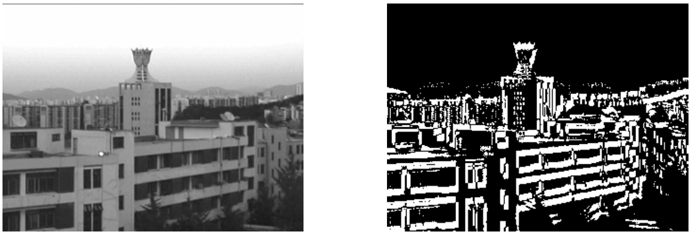


Fig. 3. Sample image frame (left) and the BE transform result (right)

For all four sub-images in an image frame, the motion vector is decided from the corresponding sub-images of the previous frame based on the phase correlation estimation using this BE image frame. For each sub-image, the largest peak amplitude location of the corresponding phase correlation surface is assigned as the LMV with the magnitude equivalent to peak amplitude. The FMV could be decided by suitably combining information from four LMVs.

The utilization of BET in the motion estimation part of the stabilization system reduces the computational fallacy as well as error like an unfeasible motion vector and is therefore particularly good at less patterned and noisy image frame.

3.2 Adaptive Motion Correction

In general, an LMV from a sub-image tends to be erroneous due to bad conditions: shading, blooming, occlusion, and noise [7]. Therefore, it should be excluded from the FMV decision process. Since the hand movement is relatively slower than the frame rate of the video camera, the FMVs of two successive frames fluctuated by a camera's jitter should be similar. Based on these properties of camera's movement, we use a simple and robust motion prediction and correction scheme. The FMV is determined by separately selecting the most maximum peak of each motion vector elements from sub-image and using (Kalman+MD) filter for selecting feasible LMVs that are close to the predicted FMV (PFMV). This is shown in Fig. 4.

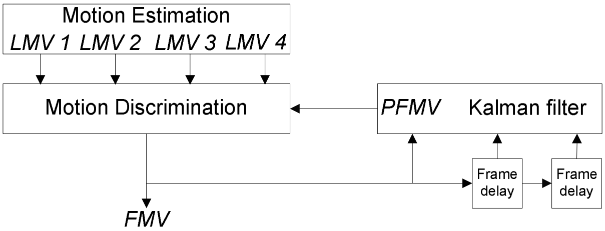


Fig. 4. Flowchart of the proposed Kalman and MD algorithms

A. Kalman filter for PFMV

The Kalman filter provides an estimate to the state of a discrete time process defined in the form of a linear dynamical system as $x(t+1) = F \times x(t) + w(t)$, with process noise $w(t)$. The Kalman filter is operated by using observations of all or some of the state variables, defined by the observation system $y(t) = H \times x(t) + v(t)$, where $v(t)$ represents measurement noise. Process and measurement noise are assumed to be independent of each other, white, having normal probability distributions, $w \sim N(0,Q)$ and $v \sim N(0,R)$. For the image stabilization system, horizontal and vertical absolute frame position state estimates by (x_1, x_2), (dx_1, dx_2) represents the corresponding velocity, and (d^2x_1, d^2x_2) represents the corresponding acceleration. The state system for the image stabilizer is constructed in the form of a constant acceleration camera model is given as

$$\begin{bmatrix} x_1(n) \\ x_2(n) \\ dx_1(n) \\ dx_2(n) \\ d^2x_1(n) \\ d^2x_2(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \\ dx_1(n-1) \\ dx_2(n-1) \\ d^2x_1(n-1) \\ d^2x_2(n-1) \end{bmatrix} + \begin{bmatrix} wx_1 \\ wx_2 \\ wdx_1 \\ wdx_2 \\ wd^2x_1 \\ wd^2x_2 \end{bmatrix}. \tag{3}$$

The Kalman filter output is obtained recursively through prediction and update stages enabling real time operation of the filter.

B. Motion Discrimination

The motion discrimination using predicted frame motion vector in the form of

$$d_n = \begin{cases} 1 \quad , & |LMV_n - PFMV| < \beta \\ 0 \quad , & |LMV_n - PFMV| > \beta \end{cases} \tag{4}$$

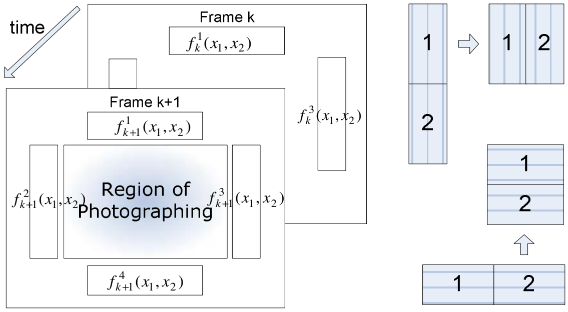
is used to filter the motion vector, and the frame motion vector is decide based on the number of feasible motion vector, which can be expressed in the form of

$$FMV = \frac{d_1 \cdot LMV_1 + d_2 \cdot LMV_2 + d_3 \cdot LMV_3 + d_4 \cdot LMV_4}{d_1 + d_2 + d_3 + d_4}, \tag{5}$$
$$d_1 + d_2 + d_3 + d_4 \neq 0.$$

The processing of MD filter discriminates between proper and improper vector from four LMVs of each sub-images using PFMV. Then calculates frame motion vector using only feasible motion vectors for accuracy of DIS. The utilization of adaptive kalman and MD filter in the motion correction part of the stabilization system overcome unfeasible motion vector from phase correlation.

4 Experimental Results

In order to further increase the speed of frame motion estimation, four sub-images designated as shown in Fig. 5(a) are used and local motion vectors are computed using BE transform based phase correlation for all sub-images. We proposed rectangular sub-images to extend Region of Photographing (ROP) and to attempt to remove moving objects in the sub-image. These sub-images are used to determine LMVs using phase correlation. For efficient FFT computation, sub-images have square shape with horizontal and vertical pixel dimensions being a power of two. Typically a sub-image size of 32×32 is preferred to reduce the computation and at the same time, to keep a sufficiently large area for a correct estimation. But the proposed sub-images are not square. Rectangular sub-image changes to square as seen in the Fig. 5(b).



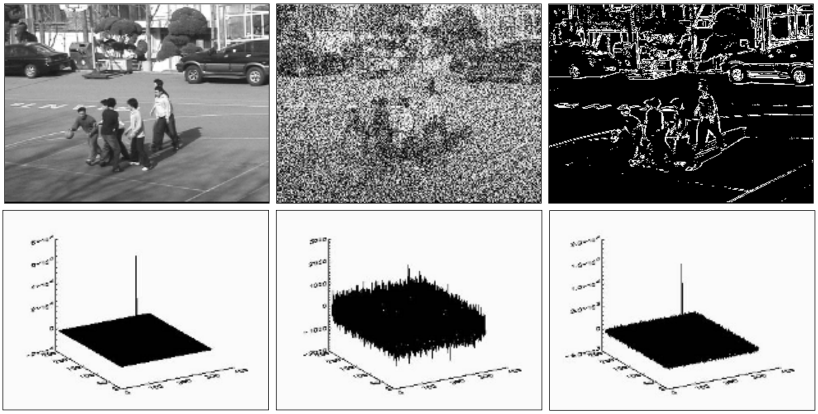
(a) (b)

Fig. 5. Location of rectangular sub-images for motion estimation. (a) Location of sub-images, (b) Change to square.

To evaluate the algorithm performance, motion estimation algorithm was used on both natural and synthetic low light image sequences. The stabilizing performance is evaluated subjectively and verified with root mean square error (RMSE) between the estimated motion vectors and the true motion vectors. The performance is evaluated using the same error measure utilized in [1][6][9]. The RMSE is given by

$$\varepsilon_{RMS} = \frac{1}{N}\left[\sum_{k=1}^{N}(\hat{x}_k-\bar{x}_k)^2+(\hat{y}_k-\bar{y}_k)^2\right]^{1/2} \qquad (6)$$

where ($\bar{x}_k$, $\bar{y}_k$) is the true values of the measurement motion vectors and ($\hat{x}_k$, $\hat{y}_k$) is the estimated motion vectors generated from the evaluated DIS algorithms. The results using the proposed stabilization algorithm for both natural and synthetic sequences are shown in Fig. 6. Test sequences are Basketball (B), Library (L), Street (S), Market (M), Café (C), Gate (G), and Phone (P). Performance comparison using the RMSE obtained by several existing stabilization algorithms is given in Table. I. Results are also compared against phase correlation based motion estimation performance as presented in [9]. For this purpose, sub-image based approaches using assignment of the local motion vector with the largest peak amplitude as the global motion vector (Sub-PC-H), assignment of the average of the two local motion vectors with the highest two peak amplitudes as the global motion vector (Sub-PC-2), assignment of the weighted average of all local motion vectors weighted by their peak



(a) Natural image (b) Synthetic noisy image (c) Proposed BET image

Fig. 6. Experimental results of the motion estimation algorithm for DIS

amplitude values (Sub-PC-W), and assignment of the sub-images one-bit transform (Sub-1BT) [6] were simulated for comparison with the proposed BE transformation based motion estimation and MD. The samples of test results with synthetic sequences are shown in Fig. 6 (b) and Fig. 6 (c).

The effectiveness of the proposed DIS algorithm in processing different images can easily be evaluated by Table I which demonstrates the stabilization results of (BE+MD) filter and the comparison filters for images degraded by noise. It can be seen from the Table I that the shake removal performance of the Sub-1BT is significantly low compared with the other operators. The outputs of the PC based motion estimation with Sub-PC-H, Sub-PC-2, and Sub-PC-W operators are almost the same, but these operators present incorrect results of motion estimation in the noisy image. The Sub-PC-BE operator exhibits much better performance than the others. Even though phase correlation based techniques provide improved motion estimation accuracy, the computational complexity is higher compared to the Sub-1BT approach, due to the requirement of Fourier domain computation. It is clearly seen that the Sub-PC-BE and MD operator successfully detects the motion vector while at the same time

Table 1. Comparison of RMSE of various DIS algorithms with the proposed BET+MD

		B	L	S	M	C	G	P
Sub-PC-H	RMSE	0.152	1.064	0.112	0.125	0.185	0.114	0.121
	(Noise)	0.521	0.638	0.452	0.598	0.463	0.479	0.513
Sub-PC-2	RMSE	0.128	0.165	0.136	0.113	0.164	0.105	0.115
	(Noise)	0.462	0.662	0.419	0.528	0.533	0.477	0.532
Sub-PC-W	RMSE	0.137	0.142	0.114	0.102	0.161	0.108	0.112
	(Noise)	0.431	0.598	0.422	0.487	0.453	0.455	0.561
Sub-1BT	RMSE	0.139	0.183	0.142	0.139	0.194	0.132	0.146
	(Noise)	0.311	0.772	0.389	0.566	0.823	0.588	0.427
Sub-PC-BET	**RMSE**	**0.125**	**0.132**	**0.098**	**0.095**	**0.128**	**0.103**	**0.095**
	(Noise)	**0.137**	**0.151**	**0.112**	**0.108**	**0.131**	**0.117**	**0.106**
Sub-PC-BET	**RMSE**	**0.123**	**0.121**	**0.102**	**0.092**	**0.116**	**0.111**	**0.091**
-MD	**(Noise)**	**0.128**	**0.132**	**0.111**	**0.0098**	**0.135**	**0.113**	**0.094**

efficiently removing the jittering in the noisy image. In addition, the proposed Sub-PC-BE and MD approach provides a reasonable robust motion estimation system for image sequence stabilization.

5 Conclusions

In this paper, we proposed a novel digital image stabilization algorithm using binary edge transform for phase correlation is proposed, which stables shaking image sequence in zoom-in highly. For the realization of the digital image stabilization, we used binary edge transform, PC estimation, and adaptive kalman + MD filter. Although the proposed algorithm is good at operating low light level (less patterned or noisy) images, it can be extended to more general applications by low cost camcorders, digital cameras, mobile phone cameras, CCTV, surveillance video systems, and television broadcasting systems.

References

1. S. Hsu, et al., “A robust digital image stabilization technique based on inverse triangle method and background detection,” *IEEE Trans. Consumer Electronics*, vol. 51, pp. 335-345, May, 2005.
2. K. Sato, et al., “Control techniques for optical image stabilizing system,” *IEEE Trans. Consumer Electronics,* vol. 39, no. 3, pp. 461-466, Aug. 1993.
3. K. Uomori, et al., “Automatic image stabilizing system by full-digital signal processing,” *IEEE Trans. Consumer Electronics*, vol. 36, no. 3, pp. 510-519, Aug. 1990.
4. J. Paik, et al., “An adaptive motion decision system for digital image stabilizer based on edge pattern matching,” *IEEE Trans. Consumer Electronics*, vol. 38, no. 3, pp. 607-616, Aug. 1992.
5. S. Ko, et al., “Fast digital image stabilizer based on gray-coded bit-plane matching,” *IEEE Trans. Consumer Electronics*, vol. 45, pp. 598-603, 1999.
6. A. Yeni, et al., “Fast digital image stabilization using one bit transform based sub-image motion estimation,” *IEEE Trans. Consumer Electronics*, vol. 49, no. 4, pp. 1320-1325, Nov. 2003.
7. H. Okuda, et al., “Optimum motion estimation algorithm for fast and robust digital image stabilization,” *IEEE Trans. Consumer Electronics*, vol. 52, no. 1, pp. 276-280, Feb. 2006.
8. S. Erturk, “Image sequence stabilization: motion vector integration (MVI) versus frame position smoothing (FPS),” *Proc. of the 2nd International Symposium on Image and Signal Processing and Analysis,* pp. 266-271, 2001.
9. S. Erturk, “Digital Image Stabilization with Sub-Image Phase Correlation Based Global Motion Estimation,” *IEEE Trans. Consumer Electronics*, vol. 49, no. 4, pp. 1320-1325, Nov. 2003.
10. Y. Matsushita, et al., “Full-frame video stabilization with motion inpainting” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1150-1163, Jul. 2006.

Contrast Enhancement Using Adaptively Modified Histogram Equalization

Hyoung-Joon Kim[1], Jong-Myung Lee[1], Jin-Aeon Lee[2], Sang-Geun Oh[2],
and Whoi-Yul Kim[1]

[1] Division of Electronics and Computer Engineering, Hanyang University,
Haengdang-Dong, Sungdong-Gu, Seoul, 133-792, Korea
{khjoon, jmlee}@vision.hanyang.ac.kr, wykim@hanyang.ac.kr
[2] Samsung Electronics,
Giheung-Eup, Yongin-Si, Gyeonggi-Do, 449-712, Korea
{jalee, stephen.oh}@samsung.com

Abstract. A new contrast enhancement method called adaptively modified histogram equalization (AMHE) is proposed as an extension of typical histogram equalization. To prevent any significant change of gray levels between the original image and the histogram equalized image, the AMHE scales the magnitudes of the probability density function of the original image before equalization. The scale factor is determined adaptively based on the mean brightness of the original image. The experimental results indicate that the proposed method not only enhances contrast effectively, but also keeps the tone of the original image.

Keywords: Contrast enhancement, image enhancement, histogram equalization.

1 Introduction

Histogram equalization (HE) is a very common method for enhancing the contrast of an image [1]. Based on the histogram or probability density function (PDF) of the original image, the image's PDF is reshaped into one with a uniform distribution property in order to enhance contrast. Although HE provides images of higher contrast, it does not necessarily provide images of higher quality [2]. That is, HE shifts the mean brightness of the image significantly and sometimes even degrades the image quality. In addition, HE causes a washed-out effect when the amplitudes of the histogram components are very high at one or several locations on the grayscale [3].

To overcome the aforementioned problems of typical HE, brightness preserving bi-histogram equalization (BBHE) has been proposed [4]. The ultimate goal of BBHE is to preserve the mean brightness of the original image while enhancing the contrast. BBHE decomposes the original image into two sub-images based on the mean brightness, and equalizes each the sub-images independently based on their corresponding PDFs. Later, equal area dualistic sub-image histogram equalization (DSIHE) [5], recursive mean-separate histogram equalization (RMSHE) [6], and minimum mean brightness error bi-histogram equalization (MMBEBHE) [7] have

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1150 – 1158, 2006.

been proposed with the same purpose of preserving the mean brightness. Although these methods can prevent significant change of gray levels and preserve the tone of the original image, they cannot sufficiently enhance the contrast of some sample images presented in Section 3. Another method, bin underflow and bin overflow (BUBO) [2], has been proposed to prevent a significant change of gray levels but still provide functionality controlling the rate of contrast enhancement in HE. Before equalization, the PDF of the original image is modified as the magnitudes of each histogram component are limited to the given upper and lower bounds. By limiting the magnitudes, a significant change of gray levels does not occur. However, since the gray levels, which the magnitudes of the histogram components are limited, are stretched linearly, as shown in Fig. 1, it is sometimes difficult to sufficiently enhance the contrast of the regions composed of pixels corresponding to those gray levels.

Based on typical HE, we propose a new and simple method called adaptively modified histogram equalization (AMHE) to enhance contrast. Similar to BUBO, AMHE modifies the PDF of the original image before equalization. Our proposed method differs by scaling the magnitudes of the PDF while preserving the original shape of the PDF rather than limiting the magnitudes in BUBO. AMHE also provides a functionality to control the rate of contrast enhancement that can be adaptively determined based on the mean brightness of the original image.

The rest of the paper is organized as follows. Section 2.1 reviews HE and BUBO. Section 2.2 describes the proposed method, and Section 2.3 presents the rule to determine adaptively the rate the contrast can be enhanced. Experimental results with comparison to different methods are presented in Section 3. Finally, the paper is concluded in Section 4.

2 Adaptively Modified Histogram Equalization

2.1 HE and BUBO

Let $\mathbf{X}=\{X(i, j)\}$ denote a given image with L gray levels, where $X(i, j)$ denotes the gray level at the location (i, j) and $X(i, j) \in [0, L\text{-}1]$. The PDF of $\mathbf{X}$ is defined as

$$p(k) = \frac{n_k}{N} \quad k = 0, 1, \ldots, L\text{-}1 \tag{1}$$

where N represents the number of pixels in $\mathbf{X}$ and n_k denotes the number of pixels with the gray level of k. Based on the PDF, its cumulative distribution function (CDF) is defined as

$$c(k) = \sum_{i=0}^{k} p(i)\,. \tag{2}$$

The mapping function of HE is defined as

$$f(k) = (L-1)c(k)\,. \tag{3}$$

The PDF of the HEed image by Eq. (3) obtains uniform distribution, which reflects the enhanced contrast. However, there is no mechanism for controlling the rate of contrast enhancement. In addition, if many pixels are concentrated on in a few gray levels, the gray levels have been changed too much and thus the resulting image looks unnatural. To overcome these shortcomings of HE, BUBO [2] modifies the original PDF with constraints as

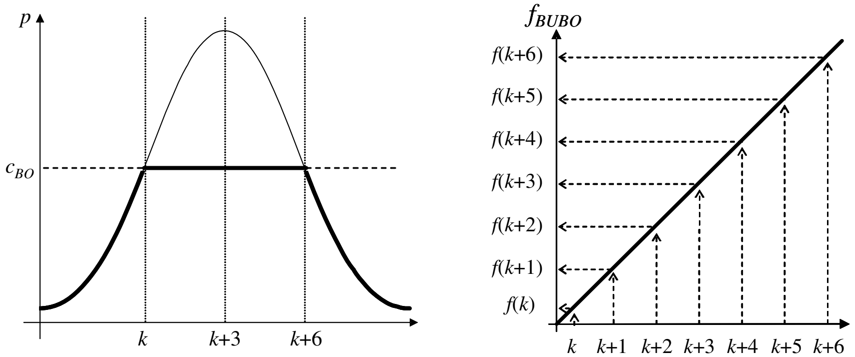
$$p_{BUBO}(k)=\begin{cases} c_{BO}, & \text{if } p(k)>c_{BO} \\ p(k), & \text{if } c_{BU}\le p(k)\le c_{BO} \\ c_{BU}, & \text{if } p(k)<c_{BU} \end{cases} \tag{4}$$

where c_{BO} and c_{BU} are thresholds for the bin underflow (BU) and bin overflow (BO), respectively, defined as

$$c_{BU}=(1-\alpha)/N \text{ and } c_{BO}=(1+\alpha)/N\,. \tag{5}$$

Here, α is a parameter to control the rate of contrast enhancement from none at $\alpha=0$ to full HE at $\alpha=\infty$.

Fig. 1 shows a simulated mapping function of BUBO. In Fig. 1 (a), plots for the original PDF, the modified PDF by Eq. (4), and c_{BO} are shown in thin solid, thick solid, and dotted lines, respectively. The probabilities greater than c_{BO} are limited to c_{BO} and thus the part of the mapping function corresponding to those gray levels has linear increment characteristics, as shown in Fig. 1 (b). This can prevent a significant change in the gray level but hinder the enhancement of contrast.



(a) Original PDF

(b) Mapping function

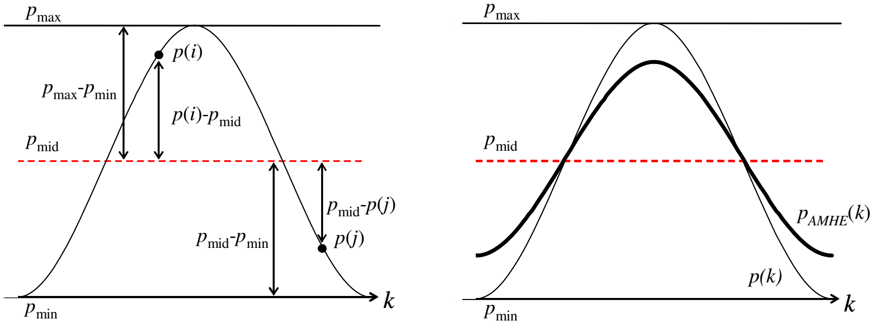
Fig. 1. The mapping function of BUBO: (a) Original PDF (thin solid), modified PDF (thick solid) by Eq. (4), and c_{BO} (dotted). (b) Mapping function between *k* and *k*+6 gray levels.

2.2 Adaptively Modified Histogram Equalization

In HE, the rate of contrast enhancement can be controlled by putting constraints on the gradient of the mapping function, i.e. the CDF [2]. Since the gradient of the CDF becomes the PDF, we can control the rate of contrast enhancement by modifying the PDF. The proposed method, adaptively modified histogram equalization (AMHE), modifies the original PDF and preserves its shape as shown in Fig. 2 by

$$p_{AMHE}(k) = \begin{cases} p_{\text{mid}} + \alpha \dfrac{(p(k) - p_{\text{mid}})^2}{p_{\max} - p_{\text{mid}}}, & \text{if } p(k) > p_{\text{mid}} \\ p_{\text{mid}} - \alpha \dfrac{(p_{\text{mid}} - p(k))^2}{p_{\text{mid}} - p_{\min}}, & otherwise \end{cases} \tag{6}$$

where $p_{\min}$ and $p_{\max}$ are the minimum and maximum values of $p(k)$, respectively, and p_{mid} is the mean value of $p_{\min}$ and $p_{\max}$. Note that $p_{AMHE}(k)$ is forcibly set to zero when negative. The rate of contrast enhancement is then determined by α, and the rule for deciding α adaptively is given in the next section.



(a) Original PDF (b) Modified PDF

Fig. 2. Modification of the PDF

For the modified PDF p_{AMHE}, the CDF c_{AMHE} is first computed by Eq. (7). Since $c_{AMHE}(L\text{-}1)$ does not become one, $\tilde{c}_{AMHE}$ is used instead as the mapping function for contrast enhancement by a simple normalization in Eq. (8).

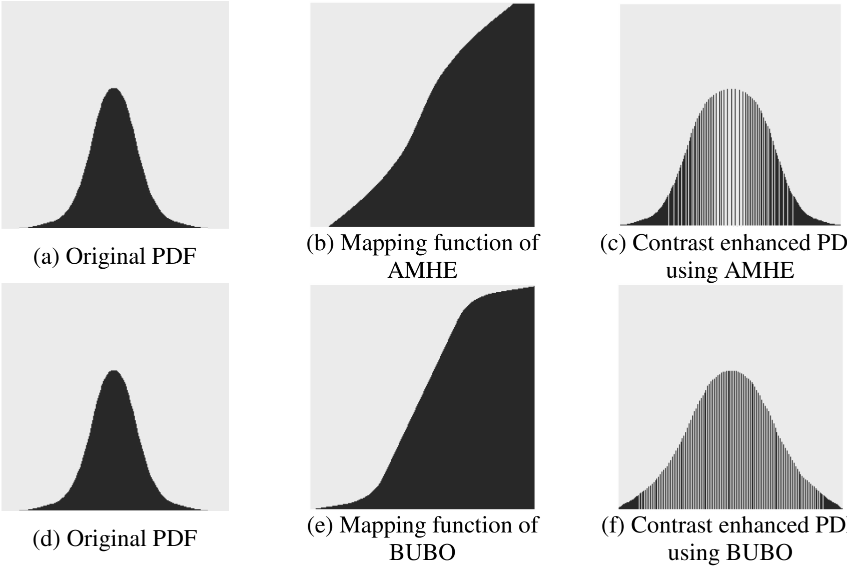
$$c_{AMHE}(k) = \sum_{i=0}^{k} p_{AMHE}(i) \tag{7}$$

$$\tilde{c}_{AMHE}(k) = \frac{c_{AMHE}(k)}{c_{AMHE}(L-1)}(L-1) \tag{8}$$

Since the shape of the original PDF is preserved, AMHE can enhance contrast with similar performance of HE while AMHE can prevent a significant change in gray level by scaling the PDF. Fig. 3 shows simulated results of AMHE and BUBO. The gray levels where the magnitudes of the histogram components are limited are linearly stretched in the enhanced result of BUBO. However, the gray levels are stretched in proportion to those probabilities in AMHE and thus the contrast of the regions in the image composed of those gray levels is more enhanced than BUBO.

2.3 Adaptive Parameter Decision

As mentioned, the strength of the modification of the PDF decided by α affects the mapping function, and thus it is important to determine the proper value of α. In our


(a) Original PDF

(b) Mapping function of AMHE

(c) Contrast enhanced PDF using AMHE

(d) Original PDF

(e) Mapping function of BUBO

(f) Contrast enhanced PDF using BUBO

Fig. 3. Simulated results of AMHE and BUBO

experiments testing various images, fine results have generally been obtained when α is set to 0.7. Nonetheless, we suggest a rule to decide α adaptively based on the mean brightness of the image by Eq. 9. Note that X_m denotes the mean brightness of the image. When the image is decomposed into two sub-images by X_m, X_{ml} indicates the mean brightness of the sub-image composed of pixels with the gray levels less than or equal to X_m while X_{mu} is the mean brightness of the other sub-image. It is obvious that X_m is larger than X_{ml} and smaller than X_{mu}.

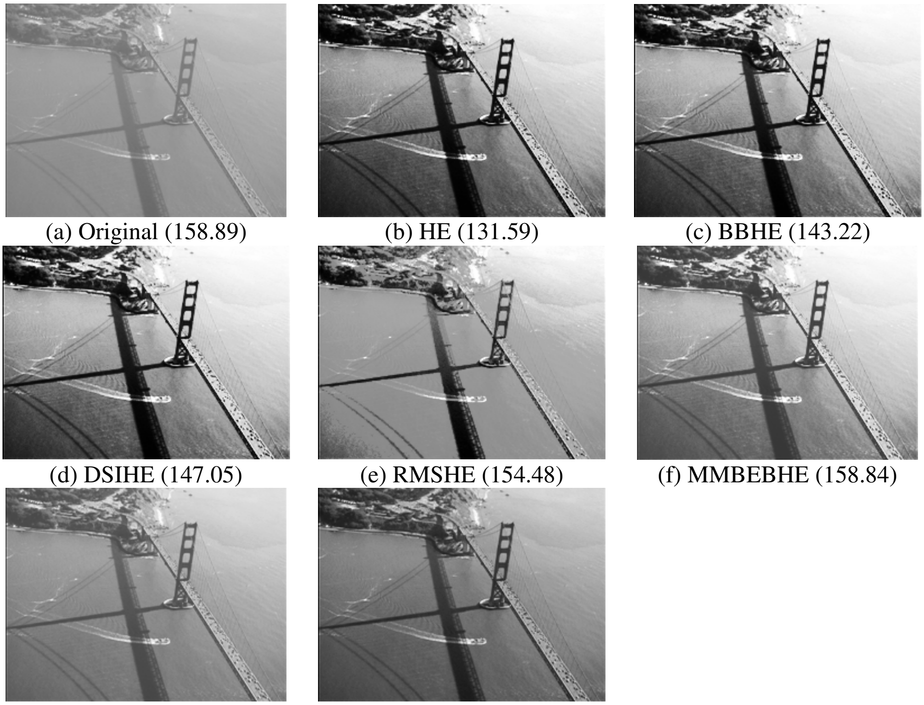
$$\begin{aligned} \alpha &= \frac{X_m - X_{ml}}{X_{mu} - X_{ml}}, \quad \text{for } 0 \le k \le X_m \\ \alpha &= \frac{X_{mu} - X_m}{X_{mu} - X_{ml}}, \quad \text{for } X_m < k \le L-1 \end{aligned} \tag{9}$$

where, $X_{ml} = \sum_{k=0}^{X_m} k \cdot p(k) \bigg/ \sum_{k=0}^{X_m} p(k)$ and $X_{mu} = \sum_{k=X_m+1}^{L-1} k \cdot p(k) \bigg/ \sum_{k=X_m+1}^{L-1} p(k)$

The motivation of the decision rule of Eq. (9) is based on the observation that in natural images many pixels are generally concentrated on mean brightness. More specifically, as the interval between X_{ml} and X_m becomes shorter, more pixels with gray levels between X_{ml} and X_m occur for the sub-image composed of pixels with the gray levels less than or equal to X_m. The same phenomenon applies to the other sub-image. In the next section, two enhanced results using this decision rule are provided.

3 Experimental Results

To demonstrate the performance of contrast enhancement, we tested the proposed method on various images gathered from the Internet as well as taken by digital

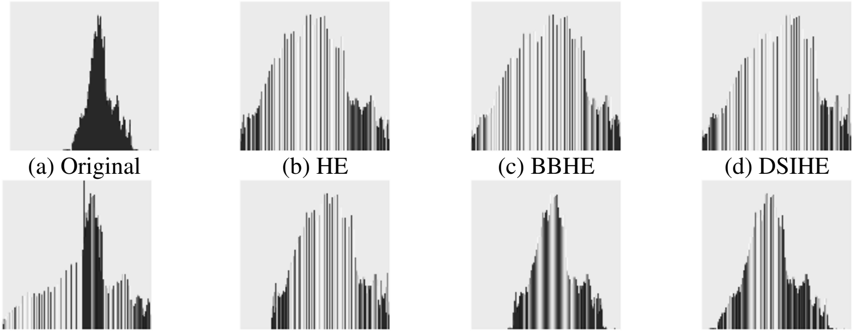


(a) Original (158.89) (b) HE (131.59) (c) BBHE (143.22)

(d) DSIHE (147.05) (e) RMSHE (154.48) (f) MMBEBHE (158.84)

(g) BUBO (148.33) (h) AMHE (122.17)

Fig. 4. Contrast enhancement results



(a) Original (b) HE (c) BBHE (d) DSIHE

(e) RMSHE (f) MMBEBHE (g) BUBO (h) AMHE

Fig. 5. The PDFs of the images in Fig. 4

cameras. Our method was also compared with different methods such as HE [1],BBHE [4], DSIHE [5], RMSHE [6], MMBEBHE [7], and BUBO [2]. Note that α in BUBO is set to 0.7 and there are two recursions in RMSHE.

Two examples are shown in Fig. 4 and Fig. 6 while the histograms of those images are shown in Fig. 5 and Fig. 7, respectively. Note that numbers in parentheses in Fig. 4 and Fig. 6 indicate the mean brightness of each image. As mentioned previously,

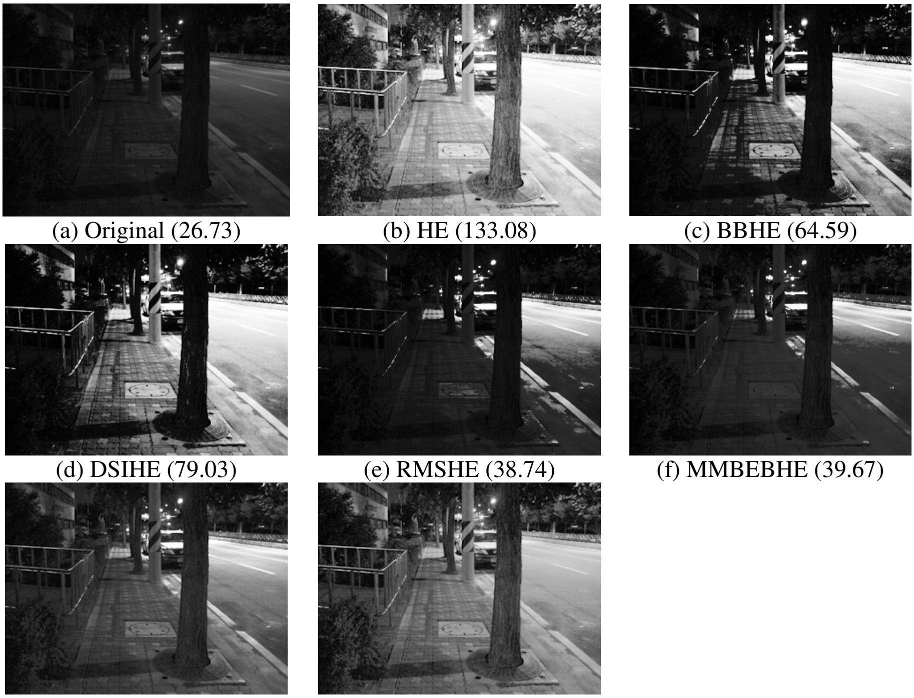

(a) Original (26.73)
(b) HE (133.08)
(c) BBHE (64.59)
(d) DSIHE (79.03)
(e) RMSHE (38.74)
(f) MMBEBHE (39.67)
(g) BUBO (66.43)
(h) AMHE (88.46)

Fig. 6. Contrast enhancement results

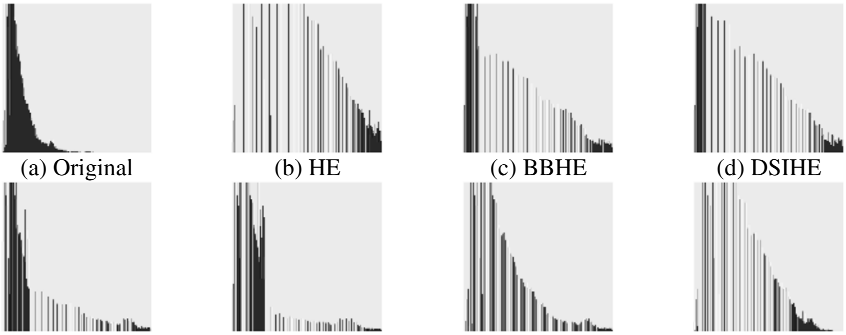

(a) Original
(b) HE
(c) BBHE
(d) DSIHE
(e) RMSHE
(f) MMBEBHE
(g) BUBO
(h) AMHE

Fig. 7. The PDFs of the images in Fig. 6

HE enhances contrast sufficiently but the enhanced images seem unnatural. BBHE, DSIHE, RMSHE, and MMBEBHE well preserve the mean brightness of the original image in comparison with HE. However, the enhanced images by them are far from satisfactory—the region of the tree is especially indistinguishable as shown in Fig. 8. BUBO and our method enhance contrast better than the rest while maintaining the

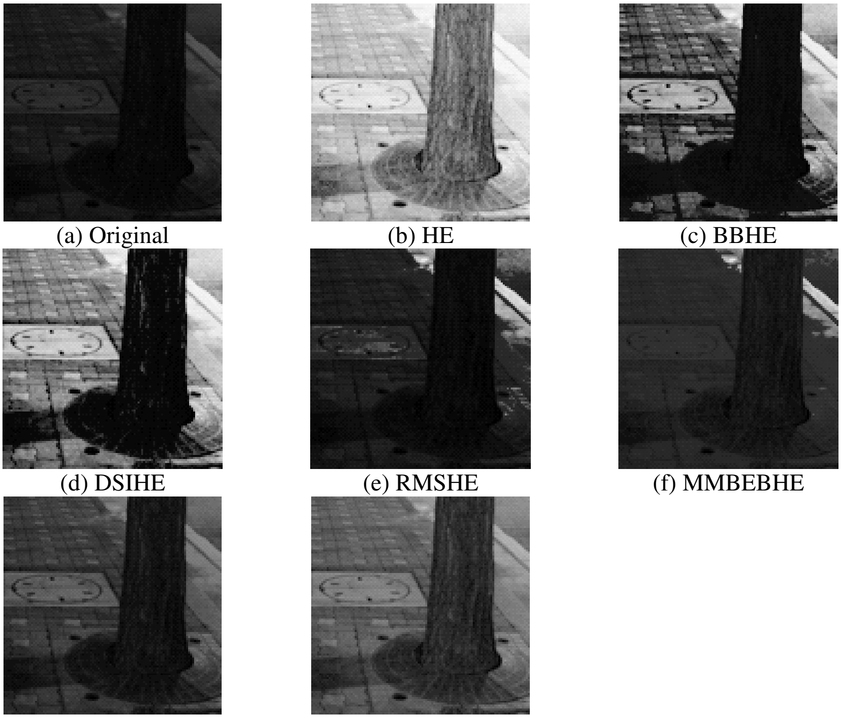

(a) Original
(b) HE
(c) BBHE
(d) DSIHE
(e) RMSHE
(f) MMBEBHE
(g) BUBO
(h) AMHE

Fig. 8. The enlarged images of Fig. 6

tone of each original image. Nevertheless, the enhanced images of our method are recognized more easily than BUBO, and the grayscale is utilized more efficiently, as shown in Fig. 5 (g) and (h).

4 Conclusions

In this paper, we present a new and simple contrast enhancement method referred to as *adaptively modified histogram equalization* (AMHE). It is an extension of typical histogram equalization (HE). AMHE modifies the probability density function (PDF) of a given image while preserving its shape and then applies HE to that modified PDF. By modifying the PDF, not only is a significant change in gray level, which often occurs in HE, prevented, but the overall contrast of the image is enhanced. Moreover, the rate of enhanced contrast can be controlled by the degree of the modification of the PDF, and is determined adaptively using the mean brightness of the image. Comparison of experimental results using different methods indicates that the proposed method not only enhances contrast effectively while keeping the tone of the original image, but most importantly it makes the enhanced image look very natural by adhering to the flavor of original image.

References

1. Gonzalez, R. C., Woods R. E.: Digital Image Processing. 2nd edn. Prentice-Hall, New Jersey (2002)
2. Yang, S. J., Oh, J. H., Park, Y. J.: Contrast Enhancement Using Histogram Equalization with Bin Underflow and Bin Overflow. International Conference on Image Processing (2003) 881-884
3. Chen, Z., Abidi, B. R., Page, D. L., Abidi, M. A.: Gray-Level Grouping (GLG): An Automatic Method for Optimized Image Contrast Enhancement—Part I: The Basic Method. IEEE Transactions on Image Processing, Vol. 15, No. 8 (2006) 2290-2302
4. Kim, Y. T.: Contrast Enhancement using Brightness Preserving Bi-Histogram Equalization. IEEE Transactions on Consumer Electronics, Vol. 43, No. 1 (1997) 1-8
5. Wang, Y., Chen, Q., Zhang, B.: Image Enhancement Based on Equal Area Dualistic Sub-Image Histogram Equalization Method. IEEE Transactions on Consumer Electronics, Vol. 45, No. 1 (1999) 68-75
6. Chen, S., Ramli, A. R.: Contrast Enhancement using Recursive Mean-Separate Histogram Equalization for Scalable Brightness Preservation. IEEE Transactions on Consumer Electronics, Vol. 49, No. 4 (2003) 1301-1309
7. Chen, S., Ramli, A. R.: Minimum Mean Brightness Error Bi-Histogram Equalization in Contrast Enhancement. IEEE Transactions on Consumer Electronics, Vol. 49, No. 4 (2003) 1310-1319

A Novel Fade Detection Algorithm on H.264/AVC Compressed Domain

Bita Damghanian[1], Mahmoud Reza Hashemi[2], and Mohammad Kazem Akbari[1]

[1] Computer Engineering & Information Technology Department,
Amirkabir University of Technology, Tehran, Iran
{damghanian, akbari}@ce.aut.ac.ir
[2] Multimedia Processing Laboratory
School of Electrical and Computer Engineering, University of Tehran, Iran
hashemi@comnete.com

Abstract. With the increasing use and availability of digital video, shot boundary detection, as a fundamental step in automatic video content analysis and retrieval, has received extensive attention in recent years. But the problem of detecting gradual transitions such as fades, especially in the compressed domain, has yet to be solved.

In this paper, a new robust compressed domain fade detection method is proposed, which is based on the luminance weighting factors available at slice level in H.264 compressed video. Simplicity and working directly in the H.264 compressed domain with high precision and recall are the main advantages of the novel algorithm.

Keywords: Content based video analysis and retrieval, shot boundary detection, fades detection, compressed domain, H.264 video coding standard, weighted prediction.

1 Introduction

To achieve automatic video content analysis, shot boundary detection is a preliminary step; hence, in recent years the research on automatic shot boundary detection techniques has exploded.

In the literature, shot is defined as the sequence of images generated by the camera from the time it begins recording to the time it stops; and according to whether the transition between shots is abrupt or not, it can be classified into two types: abrupt transition (cut) and gradual transition. Gradual transitions are further categorized as dissolve, fades, and all kinds of wipe. There are two types of fade: fade-in and fade-out. A fade-in occurs when the picture gradually appears from a black screen and a fade-out occurs when the picture information gradually disappears, leading to a black screen.

Among gradual transitions, fades are mainly used in television and movie production to emphasize time transitions. Furthermore, they can be used to separate different TV program elements such as the main show material from commercial blocks. Fade-out/in combinations can also indicate relative mood and pace between shots [1]. Therefore, detecting them is a very powerful tool for shot classification and story summarization.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1159 – 1167, 2006.

Fade detection algorithms are classified as uncompressed and compressed domain algorithms [2]. Uncompressed or pixel domain algorithms utilize information directly from the spatial video domain. These techniques are computationally demanding and time consuming; because they must decompress the coded videos to apply the method on. They are hence less efficient comparing to the compressed domain approach.

Among the international video coding standards, H.264/AVC is the most recent one and due to its superior compression performance, more video will be encoded with it in the future. Thus, it is meaningful to study fade detection techniques directly in the H.264 compressed domain.

The remainder of this paper is organized as follows. Section 2 reviews the existing techniques and related works. Section 3 presents an overview of the weighted prediction tool in the H.264 standard. The novel fade detection method is described in section 4. Simulation results are presented in section 5. Finally, the paper ends in section 6 with the conclusion and future work.

2 Related Works

Several shot boundary detection techniques have been proposed in the literature. Most existing methods focus on cut detection rather than fade or dissolve detection. They are designed based on the fact that the frames within the same shot maintain some consistency in the visual content, while the frames surrounding the shot boundaries exhibit a significant change of visual content [3]. The considered visual content can be a feature in the uncompressed domain such as object boundaries [4], color histogram [5], correlation average [6], or a value in the compressed domain like the rate of macro blocks [2], [7], motion vectors [8], and DC coefficients [9].

Meanwhile, as mentioned in [3] the gradual transitions may span over dozens of frames and the variation of the visual content between two consecutive frames may be considerably small; therefore, the identification of gradual transitions is far more difficult, and specific solutions for fade and dissolve detection is needed.

In fade-out, the first frame gradually darkens into a sequence of dark monochrome frames and in the case of a fade-in, it gradually brightens from a sequence of dark monochrome frames and these dark monochrome frames seldom appear elsewhere; as a result of this, the fade detection problem usually turns to the recognition of monochrome frames.

Monochrome frames are identified by calculating the standard deviation of each frame [10], the standard deviation together with the average intensity of each frame [3], or by calculating the correlation of an image with a constant image [6].

After locating monochrome frames, some other constraints are checked for robustness such as standard deviation of the pixels [6], the ratio between the second derivative of the variance curve and the first derivative of the mean curve [10], second derivative curve of luminance variance, and the first derivative of each frame mean [5], [1].

[5] And [11] have surveyed some other fade detection techniques which have somehow the same ideas as the above schemes.

All the methods mentioned so far, are performed in the pixel domain; and hence, are not very efficient for applications such as video indexing where we have to deal with a large amount of compressed video content. Few other methods have been proposed for the MPEG compressed domain. Most of them have a moderate accuracy and precision. For instance, [12] declares a fade-in, if the number of positive residual DC coefficients in P frames, exceeds a certain percentage of the total number of non-zero DC coefficients consistently over several consecutive frames. Simplicity is the advantage of this technique, because it only uses entropy decoding; but on the other hand, it does not have a high accuracy and can not locate the exact boundaries of fades. [10] Has an approximation of its pixel domain solution for the MPEG-2 and H.263 compressed domain. It approximates the mean and the variance using a DC-estimation of each picture. Results are distorted due to the limitations of variance calculation.

There are even fewer methods for the H.264 compressed domain. One of the few existing techniques has been proposed in [13], which at first, uses the intra prediction mode histogram to locate potential GOPs (Group of Pictures), where shot transitions occur with great probability. It then utilizes more inter prediction modes and Hidden Markov Models to segment the video sequence. This method is able to fairly detect cuts and dissolves, but not fades.

To summarize, there is a real need for a robust fade detection algorithm in the H.264 compressed video. In this paper a new three step algorithm for fade detection is proposed.

3 Overview of the Weighted Prediction Tool in the H.264 Standard

Weighted Prediction (WP) is a newly adopted concept in the prediction component of the H.264 video coding standard. It is available in the Main and Extended profiles.

Unlike motion estimation which looks for the displacement of moving objects, weighted prediction is interested in changes in light or color.

The H.264 encoder determines a single weighting factor for each color component of each reference picture index and encodes it in the bit stream slice header when the weighted prediction explicit mode is selected. Thus, each color component of the reference frame is multiplied by the weighting factor and the motion compensation is performed on the weighted reference. This results in a significant improvement in the performance of motion estimation and a lower prediction error. More details can be found in [14].

The novel fade detection method proposed here, exploits this explicit weighted prediction feature of the H.264 video coding standard.

4 The Proposed Method

In this paper a new three step algorithm for fade detection is proposed. The proposed method uses the Luminance WP factors, available at slice level of the H.264 compressed video, to detect fade transitions and their exact boundaries. The first step in the proposed algorithm detects the potential fades and their types (fade-in, fade-out).

False alarms are identified and removed in the second and third steps. The third step is also responsible for locating the exact boundaries of fades in the given video sequence.

Some of the key advantages of the proposed method over existing techniques are as follows:

1. It works directly in the H.264 compressed domain and only the slice headers are decoded; hence, improving the efficiency.
2. The proposed algorithm is very simple and requires inexpensive computation.
3. It is able to find the exact boundaries of fades and their types (fade-in, fade-out).

The three steps are described in more details in the following sections.

4.1 First Step: Detecting Potential Fades and Their Types

According to [15], each P or B slice is at least predicted from one reference frame; which is the previous I or P frame for P slices and the next I or P frame for B slices.

Hereafter, we refer to the *Luminance Weighted Prediction Factor* of P and B slices relative to these references, as PLWP and BLWP factors, respectively.

These factors can be considered as the ratio of the overall luminance of the current slice with respect to the reference frame; hence, for PLWP factor, one of the following conditions is plausible:

1. If this ratio is one, there is no change of light between the current and the reference frame. This is usually the case inside a shot where there is no considerable change of light, and is also true in cut and cross-dissolve transitions where the two shots have the same overall luminance or in object/camera motion where there is no change of light.
2. If this ratio is greater than one, the current frame is brighter than the reference frame. This may potentially represent a fade-in, where there is an overall increase of luminance.
3. If this ratio is less than one, the current frame is darker than the reference frame. This may potentially represent a fade-out, where there is an overall decrease of luminance.

Based on the above statement, in the first step of the proposed method, the frames with PLWP/BLWP factors not equal to one are selected as fade candidates.

If the corresponding PLWP factors are greater than one or if the corresponding BLWP factors are less than one, they are probable fade-ins, otherwise they are probable fade-outs. (As noted earlier, P and B slices are referenced in opposite direction; hence, when PLWP factor is greater than one, the BLWP factor is less than one and vice versa.)

4.2 Second Step: Removing False Alarms

It should be noted that not all the cases where the PLWP/BLWP factors are not equal to one, represent a fade. For instance, in an abrupt transition between two shots with different global luminance, or when there is a sudden flash light inside a shot, the PLWP/BLWP factors are not equal to one, for two or three consecutive frames. In other words, using the simple method in the previous step is not enough and may result in false alarms.

The second step is designed based on the definition of a fade. Accordingly, in fades, the overall luminance changes over N consecutive frames; thus, if the PLWP and BLWP factors are not equal to one for less than N consecutive frames, they will be omitted from the list of fade candidates. The value for N can be varied from 10 to about 150 frames. Therefore, we limit N to 10, in order not to miss any real fade.

4.3 Third Step: Removing False Alarms and Locating the Exact Fade Boundaries

Additive dissolve, slow camera motion, or slow object movement may also introduce some false alarms. They may change the luminance of consecutive frames and make some long term variations in PLWP and BLWP factors.

To make our algorithm robust to these false alarms and to find the exact fade boundaries, the third step of the proposed algorithm is designed based on the rule derived from the mathematical model of fades. We show this analytically for fade-in, but it can be similarly shown for fade-out, as well.

According to [16], if $G(x; y)$ is a grey scale sequence and N is the length of the fade transition, a fade-in sequence (F) is modeled by:

$$F(x; y; t) = G(x; y) \times \frac{t}{N}. \tag{1}$$

Thus, the fade-in transition frames are calculated as follows:

*F (0) =G*0, F (1) = G*1/N, F (2) =G*2/N, F (3) = G*3/N*, etc.

Obviously the ratio of the third frame luminance to the second frame luminance (*F (2) / F (1)*) will be 2 and the ratio of the forth frame luminance to the third frame luminance (*F (3) / F (2)*) will be 3/2, etc.

If the video sequence is coded in the IPPP format, the PLWP factor shows the ratio of the overall luminance of the current slice to the overall luminance of the previous frame; therefore, the PLWP factors will be approximately: 2, 3/2, 4/3, 5/4, 6/5, etc.

In other video coding formats such as IBP or IBBP the reference is not the previous frame. For this reason, these exact values are not obtained, but PLWP factors are still in descending order. For instance, in IBP format the reference frame of each P frame will be the frame before the previous one. Thus, the PLWP factors are obtained from (*F (4) / F (2), F (6) / F (4)...*). Hence, the PLWP factors will be approximately: 4/2, 6/4, 8/6, etc.

Consequently, we can conclude the third step of the algorithm: Among the frames selected in the previous steps as fade candidates, those that have PLWP factors in a descending order are accepted as such. The first and the last points of PLWP factors decrease are the start and the end points of the fade transition, respectively.

4.4 Summarizing the Three Steps

The simplified flowchart of the novel method is shown in figure 1. Reducing to essentials, only the PLWP factors are considered in the flow diagram.

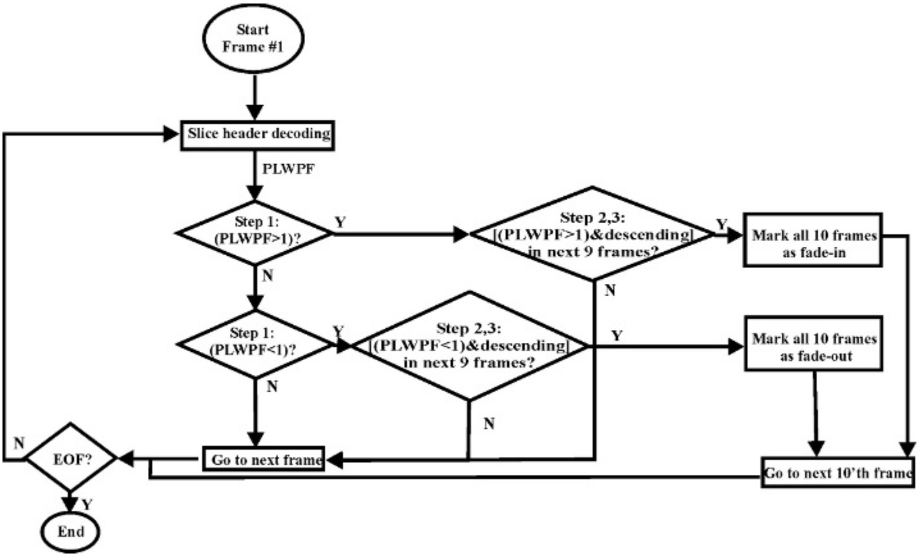


Fig. 1. Simplified flowchart of the novel method. Only the PLWP Factors are considered in the flow diagram.

At this stage, the complete fade detection method can be summarized as follows: If the PLWP factors change in a descending order from a value greater than one for at least 10 consecutive frames, a fade-in is detected. In the same manner, if the PLWP factors change in a descending order from a value less than one for at least 10 consecutive frames, a fade-out is detected. In both fade types, the first and the last point of decrease are the start and the end points of fade. Figure 2 shows clearly the fades detected by the proposed algorithm based on the PLWP factors for one sample sequence.

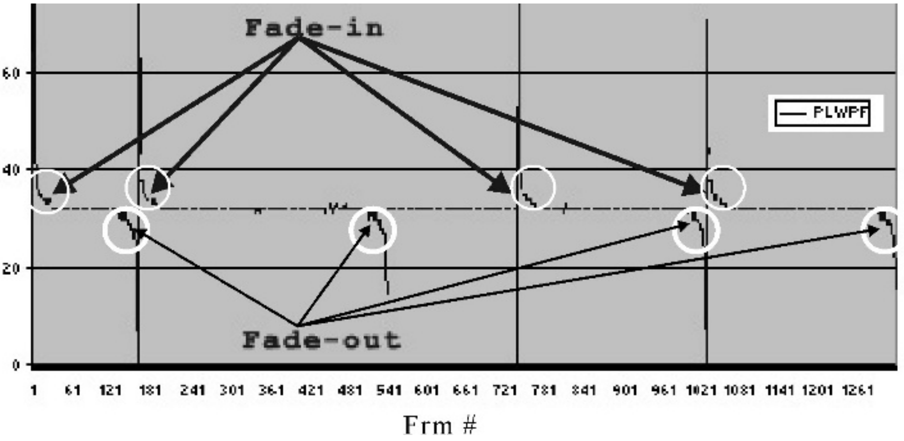


Fig. 2. PLWP factors of one sample sequence, scaled by 32. The detected fades are marked. Object motion caused some variations in the PLWP factors (e.g. frames 448 to 477). These variations were correctly removed in the third step.

In the presence of B frames, the BLWP factors must also change in an ascending order from a value less than one for at least 10 consecutive frames to declare a fade-in. In the same manner, the BLWP factors must also change in an ascending order from a value greater than one to declare a fade-out.

5 Simulation Results

To evaluate the proposed technique, it has been simulated and applied to five H.264 encoded sequences (resolution: QCIF, CIF, 30 frames/sec) with three GOP formats including IPPP, IBP, and IBBP. The test sequences contained a total of 8378 frames including 27 fades with varying length (10 to 170 frames), 17 cuts, and 13 dissolves.

The test sequences and transitions have been selected to not only provide a variety of real case transition scenarios, but also to provide various disturbance factors, such as various lighting conditions, camera motion and activity (zoom, tilt, and pan), object motion with various speeds, cuts, and dissolves, to fairly evaluate the effectiveness, and the robustness of the proposed scheme.

Experimental results show that the new method can not distinguish some special cross-dissolves from fades. These special cross-dissolves are those between two shots with very different luminance and from the observer point of view, they are very similar to fades.

To remove these false alarms, another step can be added to the proposed algorithm to recognize whether the first frame of the detected fade-in and the last frame of the detected fade-out are black or not. If they are not black, they will be removed from the list of fades.

Recall, precision, cover-recall, and cover-precision are the main measures, used to evaluate the performance of the novel method. They are defined in TRECVID-2005[17]. Accordingly, recall is the percentage of true fades detected, and precision is the percentage of non false positives in the set of detection. Cover-precision and cover-recall are specific forms of precision and recall that evaluate the detector ability to accurately locate the start and the end of a fade. Each metric is averaged over each GOP format and the result is summarized in table1.

Since at the moment no other fade detection method working directly in the H.264 compressed domain, is reported in the literature; we haven't been able to compare our results with any other technique. But, current experimental results indicate that the novel method, presented in this paper, has a high precision and recall and can locate the exact boundaries with a high cover-precision and cover-recall.

Table 1. Fade detection results, where N_c is the number of correctly detected fades, N_m is the number of fades missed by the proposed method, N_f is the number of falsely detected fades, C-P is cover-precision, and C-R is cover-recall

GOPformat	N_c	N_m	N_f	Precision	Recall	C-P	C-R
IPPP	25	2	0	100	92.59	97.95	99.25
IBP	27	0	2	93.10	100	96.89	99.12
IBBP	27	0	3	90	100	93.86	97.72

6 Conclusion and Future Work

In this paper, a new robust H.264 compressed domain fade detection method is proposed. The new technique is not only working directly in the H.264 compressed domain with low complexity, but also has a high precision and recall. The new algorithm was evaluated in terms of recall, precision, cover-recall, and cover-precision.

Our future work is to further improve the precision by recognizing whether the first frame of the detected fade-in and the last frame of the detected fade-out are black or not. If they are not black, they will be removed from the list of fades.

In addition, it is desired to stretch the new technique to operate on sequences where the weighted prediction is not used.

Acknowledgments. This work was supported in part by Iran Telecommunication Research Center (ITRC).

References

1. B.T. Truong, C. Dorai, S. Venkatesh: Improved Fade and Dissolve Detection for Reliable Video Segmentation. In: Proc. International Conference on Image Processing, Vancouver, BC, Canada, Vol. 3(2000) 961-964
2. J. Calic and E. Izquierdo: Towards Real-Time Shot Detection in the MPEG-Compressed Domain. In: Proc. of the Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Finland (2001) 1390–1399
3. W. Zheng, J. Yuan, H. Wang, F. Lin, B. Zhang: A Novel Shot Boundary Detection Framework. In: Proc. of VCIP: Visual Communications and Image Processing, Vol.5960, Bellingham, WA (2005) 410-420
4. P.N. Hashimah, L. Gao, R. Qahwaji, J. Jiang: An Improved Algorithm for Shot Cut Detection. In: Proc. of VCIP: Visual Communications and Image Processing, Vol. 5960, Bellingham, WA (2005) 1534-1541
5. B.T. Truong, Ch. Dorai, S. Venkatesh: New Enhancements to Cut, Fade, and Dissolve Detection Processes in Video Segmentation. In: Proc. of the 8th ACM international conference on Multimedia, Marina del Rey, California, United States (2000) 219 - 227
6. S. Porter, M. Mirmehdi, B. Thomas: Detection and Classification of Shot Transitions. In: British Machine Vision Conference (BMVC) (2001) 73-82
7. J. Nang, S. Hong, Y. Ihm: An Efficient Video Segmentation Scheme for MPEG Video Stream using Macroblock Information. In: Proc. of the 7th ACM international conference on Multimedia, Orlando, Florida, United States (1999) 23 - 26
8. W.S. Chau, Oscar C. Au, T.W. Chanc, T.S. Chongd: Efficient and Reliable Scene Change Detection in MPEG Compressed Video. In: Proc. of SPIE: Visual Communications and Image Processing, Vol. 5960, Bellingham, WA (2005) 1542-1549
9. K. Hoashi, M. Sugano, M. Naito, K. Matsumoto, F. Sugaya, Y. Nakajima: Shot Boundary Determination on MPEG Compressed Domain and Story Segmentation Experiments for TRECVID 2004. In: TREC Video Retrieval Evaluation Forum, JAPAN (2004)
10. W.A.C. Fernando, C.N. Canagarajah, D.R. Bull: Fade and Dissolve Detection in Uncompressed and Compressed Video Sequences. In: Proc. International Conference on Image Processing (ICIP), Kobe, Japan, Vol. 3(1999) 299-303

11. R. Lienhart: Reliable Transition Detection in Videos: A Survey and Practitioner's Guide. In: International Journal of Image and Graphics, Vol.1 (2001) 469-486
12. Divakaran, Ajay, Ito, Hiroshi, Sun, Huifang, Poon, Tommy: Fade-in/out Scene Change Detection in the MPEG-1/2/4 Compressed Video Domain. In: Proc. SPIE Storage and Retrieval for Media Databases, Vol. 3972 (2000) 518-522
13. Y. Liu, W. Zeng, W. Wang, W. Gao: A Novel Compressed Domain Shot Segmentation Algorithm on H.264/AVC. In: ICIP: International Conference on Image Processing, Vol.4 (2004) 2235-2238
14. J. M. Boyce: Weighted Prediction in the H.264/MPEG AVC Video Coding Standard. In: Proc. of the 2004 International Symposium on Circuits and Systems, ISCAS '04, Vol.3 (2004) 789-92
15. L. E. G. Richardson: H.264 and MPEG-4 Video Compression: Video Coding for Next Generation. John Wiley & Sons (2003)
16. Z. Cernekova, C. Nikou, I. Pitas: Entropy Metrics Used for Video Summarization. In: Proc. 18th spring conference on Computer graphics, New York, USA (2002) 73 - 82
17. A. Smeaton, P. Over: TRECVID-2005: Shot Boundary Detection Task Overview. (2005)

A Representation of WOS Filters Through Support Vector Machine

W.C. Chen and J.H. Jeng

Department of Information Engineering, I-Shou University,
Kaohsiung County, Taiwan
d890305d@stmail.isu.edu.tw, jjeng@isu.edu.tw

Abstract. A weighted order statistic filter (WOSF) generates a Boolean function by the specified parameters. However, WOSF with different set of parameters may generate the same Boolean function. In this paper, one proposes an alternative representation for WOS filters to characterize WOSF. This method is based on the maximal margin classification of SVM. From a truth table generated by a linearly separable Boolean function, one takes the inputs and outputs to form a training set. By SVM, it generates a maximal margin hyperplane. The hyperplane has an optimal normal vector and optimal bias, and defines a discriminant function. The discriminant function decides the category of training data and can be used to represent WOSF. In other words, one merely utilizes a normal vector and bias to represent all WOS filters with same output but different weight vectors and threshold values.

Keywords: linearly separable, Boolean function, maximal margin, normal vector, WOSF, SVM.

1 Introduction

Stack filters [1] [2] are the filters based on threshold logics. It is known that any stack filter can be characterized by a Boolean function and with nonnegative weights and threshold values, these stack filters are called weighted order statistic filters (WOSF) [3].

The WOSF, including median filters, weighted median filters (WMF), and order statistic filters (OSF), belong to the class of nonlinear filters. Because the implementation of WOS filters is based on their statistical and deterministic properties, they have good performance for edge preservation and noise suppression. Therefore they can be applied in digital signal processing such as noise cancellation, image reconstruction, edge enhancement, and texture analysis [4].

Some scholars have presented several papers about the representation of WOS filters [5-8], as Pertti Koivisto designed WOS filters by the characterization of training-based optimization, and C. E. Savin used linearly separable stack-like architecture to design WOS filters. Their representations of WOS filters are based on the pair of a weight vector and a threshold value. Various weight vectors or threshold values can represent different WOS filters.

In this paper, one proposes a different way to represent WOS filters. This sense is based on the characterization of maximal margin classification on Support Vector

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1168 – 1176, 2006.

Machine (SVM). From a truth table generated by a linearly separable Boolean function, one takes the inputs and outputs to form a training set. Based on SVM training, one can generate a maximal margin hyperplane. This hyperplane has an optimal normal vector and an optimal bias. By means of the normal vector and bias, one defines a discriminant function. The sign of the discriminant function can represent the WOSF. In other words, one merely utilizes a normal vector and a bias that can represent all WOS filters with the same output but different weight vectors and threshold values. An example of three variables is given to illustrate and verify the proposed characterization method.

2 WOS Filters

For an n-dimension input vector $\vec{x}_i \in \{0,1\}^n$, the number of nonzero components of $\vec{x}_i$ is called Hamming weight, which is denoted by $|\vec{x}_i|$. Let $f_r(\vec{x}_i)$ be a function with domain $\{0,1\}^n$ and range $\{0,1\}$ in which parameter r, $0 \le r \le n+1$ is referred to as an order. The function as follows:

$$f_r(\vec{x}_i) = \begin{cases} 1 , & \text{if } |\vec{x}_i| \ge r \\ 0 , & \text{else} \end{cases} \tag{1}$$

For each fixed r in (1), the function $f_r(\vec{x}_i)$ produces a pack of outputs. When r is changed, the function will produce another pack of outputs. Therefore the function $f_r(\vec{x}_i)$ can produce $n+1$ packs of outputs. Such function $f_r(\vec{x}_i)$ is called an order statistic filter (OSF).

For the case of $n=3$, let $\vec{x}_i = (x_1, x_2, x_3) \in \{0,1\}^3$, the order may be $r = 0,1,2,3 \text{ or } 4$. It means that the functions $f_0(\vec{x}_i)$, $f_1(\vec{x}_i)$, $f_2(\vec{x}_i)$, $f_3(\vec{x}_i)$ and $f_4(\vec{x}_i)$ can produce 5 packs of outputs. The outputs of these 5 OSF are given in table 1. In table 1, each function has the same inputs. When the order r varies, the outputs are changed. For $r=2$, the filter $f_2(\vec{x}_i)$ has outputs $(0\ 0\ 0\ 1\ 0\ 1\ 1\ 1)$.

An OSF $f_r(\vec{x}_i)$ can be extended to a WOSF, weighted OSF $f_{\Omega,r'}(\vec{x}_i)$. The quantity Ω is a weight vector and r' is a threshold value, $0 \le r' \le n'+1$. Given a weight vector $\Omega = (\omega_1\ \omega_2\ \ldots\ldots\ldots\ \omega_n)$, $\omega_i \in R$, the parameter n' is the inner product of Ω and input vector $\vec{x}_i$, i.e., $n' = \sum \omega_i x_i$. The parameter r' is a threshold value lying between 0 and $n'+1$. The function $f_{\Omega,r'}(\vec{x}_i)$ can be defined as follows:

$$f_{\Omega,r'}(\vec{x}_i) = \begin{cases} 1 , & \text{if } \sum \omega_i x_i \ge r' \\ 0 , & \text{else} \end{cases} \tag{2}$$

For each fixed r' in (2), the function $f_{\Omega,r'}(\vec{x}_i)$ produces a pack of outputs. Altering r', the function will produce another pack of outputs. Therefore the function $f_{\Omega,r'}(\vec{x}_i)$

can produce n'+1 packs of outputs. Such function $f_{\Omega,r'}(\vec{x}_i)$ is called a weight order statistic filter (WOSF).

Let $n = 3$ and the weight vector $\Omega = (1\ 3\ 1)$. The threshold values may be $r' = 0, 1, 2, 3, 4, 5$ and 6. It means that these functions $f_{\Omega,0}(\vec{x}_i)$, $f_{\Omega,1}(\vec{x}_i)$,........, $f_{\Omega,5}(\vec{x}_i)$, and $f_{\Omega,6}(\vec{x}_i)$ can produce 7 packs of outputs. The outputs of 7 WOSF are given in table 2. In table 2, each function has the same inputs and fixed weight vector Ω. When the threshold r' varies, the filter and outputs are changed. For $r' = 2$, the filter $f_{\Omega,2}(\vec{x}_i)$ has outputs $(0\ 0\ 1\ 1\ 0\ 1\ 1\ 1)$. It is clear that when the weight vector is $\Omega = (1\ 1\ ....1)$, the WOS filter is just an OS filter.

A Boolean function $f(\vec{x}_i)$ is a function having inputs $\vec{x}_i \in \{0,1\}^n$ and outputs $y_i \in \{0,1\}$. Each Boolean function can generate a truth table. A Boolean function is linearly separable if there exists a vector $\vec{\alpha} = (\alpha_1,........,\alpha_n) \in R^n$ and a threshold value $\theta \in R$, such that

$$f(\vec{x}_i) = \begin{cases} 1\ , & \text{if} \quad \sum \alpha_i x_i \geq \theta \\ 0\ , & \text{else} \end{cases} \tag{3}$$

In comparison to equation (2), every WOS filter is obviously a linearly separable Boolean function.

3 WOS Filters Induced from SVM

On learning theory, the support vector machine (SVM) is a supervised learning and can be used as a classifier or regressor. When it is a classifier, the discriminant function is used to classify the training data. If the training data is linearly separable, the most important character of SVM is that the separable hyperplane has maximum margin property. It means that the distance from the training data to the hyperplane can be maximized. Moreover, only few of the training data are needed to represent the discriminant function, i.e., the hyperplane. These representative data are referred to as support vectors.

Given $X \subseteq R^n$, $Y = \{1,-1\}$, and the training set $S = \{(\vec{x}_i, y_i)\}_{i=1}^{l} \subseteq X \times Y$, the vector $\vec{x}_i$ is as data information and y_i is the 2-class category. If the training set S is linearly separable, then there exists a hyperplane $H_{\vec{w},b}$

$$H_{\vec{w},b} = \{\vec{x}_i \in \Re^n : f_{\vec{w},b}(\vec{x}_i) =< \vec{w}, \vec{x}_i > +b = 0,\ i = 1.....l\,\}$$

such that this hyperplane can correctly classify the training set.

Given a pair $(\vec{w}, b)$, $\vec{w} \in R^n$ and $b \in R$, define $f_{\vec{w},b}(\vec{x}_i)$ by

$$f_{\vec{w},b}(\vec{x}_i) =< \vec{w}, \vec{x}_i > +b$$

The normalized function $g_{\vec{w},b}$ of $f_{\vec{w},b}$ is given by

$$g_{\vec{w},b}(\vec{x}_i) = \| \vec{w} \|^{-1} f_{\vec{w},b}(\vec{x}_i) = < \| \vec{w} \|^{-1} \vec{w}, \vec{x}_i > + \| \vec{w} \|^{-1} b \ , \ \vec{x}_i \in R^n$$

The functional margin $\mu_s(\vec{w}, b)$ and the geometric margin $\eta_s(\vec{w}, b)$ of a training set S that is defined as

$$\mu_s(\vec{w}, b) = \min_{i=1}^{l} y_i \cdot [< \vec{w}, \vec{x}_i > + b] = \min_{i=1}^{l} y_i \cdot f_{\vec{w},b}(\vec{x}_i)$$

$$\eta_s(\vec{w}, b) = \min_{i=1}^{l} y_i \cdot [< \|\vec{w}\|^{-1} \vec{w}, \vec{x}_i > + \|\vec{w}\|^{-1} b] = \min_{i=1}^{l} y_i \cdot g_{\vec{w},b}(\vec{x}_i)$$

The margin γ_s of the training set S is the maximum geometric margin of all hyperplanes defined as

$$\gamma_s = \max_{\vec{w},b} \min_{i=1}^{l} [y_i \cdot g_{\vec{w},b}(\vec{x}_i)] = \max_{\vec{w},b} \min_{i=1}^{l} [y_i \cdot (< \|\vec{w}\|^{-1} \vec{w}, \vec{x}_i > + \|\vec{w}\|^{-1} b)]$$

The hyperplane with the maximum geometric margin is called maximal margin hyperplane or optimal hyperplane.

For a linearly separable training set S, the margin of training set equals to the inverse of the Euclidean norm of $\vec{w}$. Hence one considers the following primal optimization problem:

$$\begin{array}{ll} \text{minimize} & 2^{-1} \vec{w}^T \vec{w} \\ \text{subject to} & y_i \cdot [< \vec{w}, \vec{x}_i > + b] \geq 1, \quad \forall \ i \in l \end{array} \tag{4}$$

Suppose the pair $(\vec{w}^*, b^*)$ is the solution above primal optimization problem, the maximal margin hyperplane can be written as

$$H_{\vec{w}^*,b^*} = \{ \vec{x}_i \in \Re^n : f_{\vec{w}^*,b^*}(\vec{x}_i) = < \vec{w}^*, \vec{x}_i > + b^* = 0 \}$$

where $\vec{w}^* \in R^n$, $b^* \in R$ and $\gamma_s = \|\vec{w}^*\|^{-1}$.

To solve the optimization problem, practically, one uses Lagrangian Theorem to convert (4) to the dual optimization problem [9].

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{l} z_i - 2^{-1} \sum_{i=1}^{l} \sum_{j=1}^{l} z_i z_j y_i y_j < \vec{x}_i, \vec{x}_j > \\ \text{subject to} & \sum_{i=1}^{l} z_i y_i = 0 \text{ and } z_i \geq 0, \ \forall \ i \in l \end{array} \tag{5}$$

Suppose parameter z^* is the solution of dual optimization problem, one defines the index set I_{SV} for support vectors, $I_{SV} = \{ i \in l : z_i^* > 0 \}$. The optimal normal vector $\vec{w}^*$ can be written as

$$\vec{w}^* = \sum_{i=1}^{l} z_i^* y_i \vec{x}_i = \sum_{i \in I_{SV}} z_i^* y_i \vec{x}_i$$

By means of the Karush-Kuhn-Tuker (KKT) conditions [9]

$$\begin{aligned} &z_i^*[y_i < \vec{w}^*, \vec{x}_i > + y_i b^* - 1] = 0, \\ &y_i[< \vec{w}^*, \vec{x}_i > + b^*] - 1 \geq 0, \\ &z_i^* \geq 0. \end{aligned} \tag{6}$$

the optimal bias b^* is written as

$$b^* = y_k - < \vec{w}^*, \vec{x}_k >= y_k - < \sum_{i \in I_{SV}} z_i^* y_i \vec{x}_i, \vec{x}_k >= y_k - \sum_{i \in I_{SV}} z_i^* y_i < \vec{x}_i, \vec{x}_k >$$

where k is an arbitrary element in I_{SV}.

Hence the discriminant function is defined as

$$f_{\vec{w}^*, b^*}(\vec{x}) =< \vec{w}^*, \vec{x} > + b^* = \sum_{i \in I_{SV}} z_i^* y_i < \vec{x}_i, \vec{x} > + b^* \tag{7}$$

For any index $i \in I_{SV}$, the vector $\vec{x}_i$ is called a support vector, in which the corresponding Lagrange multipliers $z_i^* > 0$. The support vectors are the nearest data points away from hyperplane. Also, they are the points with the most difficulty to be classified.

Based on the KKT conditions, one has the following equation

$$y_i \cdot [< \vec{w}^*, \vec{x}_i > + b^*] = 1 \tag{8}$$

This equation reveals that when category $y_i = +1$, the quantity $< \vec{w}^*, \vec{x}_i > + b^* = +1$ or when $y_i = -1$, $< \vec{w}^*, \vec{x}_i > + b^* = -1$. Because the optimal hyperplane $H_{\vec{w}^*, b^*}$ can correctly separate the training set, hence one has

$$y_i = \begin{cases} +1, & \text{if } f_{\vec{w}^*, b^*} > 0 \\ -1, & \text{else} \end{cases} \tag{9}$$

By the equation (7), (8) and (9), the sign of discriminant function $\operatorname{sgn}(f_{\vec{w}^*, b^*}(x))$ is defined as follows:

$$\operatorname{sgn}(f_{\vec{w}^*, b^*}(\vec{x}_i)) = \begin{cases} +1, & \text{if } < \vec{w}^*, \vec{x}_i > + b^* > 0 \\ -1, & \text{else} \end{cases} \tag{10}$$

Since the sign of discriminant function can decide the data classified to “+1” category or “-1” category, compared to equation (3), this discriminant function can also represent a linearly separable Boolean function. One now takes the inputs and the outputs to form a training set $S = \{(\vec{x}_i, y_i)\}_{i=1}^{l}$ from a truth table generated by a linearly separable Boolean function. Based on SVM training, one can generate a maximal margin hyperplane $H_{\vec{w}^*, b^*}$. This hyperplane has an optimal normal vector $\vec{w}^*$ and an optimal bias b^*, from which one can define a discriminant function

$f_{\vec{w}^*,b^*}(\vec{x}_i) =< \vec{w}^*, \vec{x}_i > +b^*$. Because the truth table also represents lots of WOSF with the same output, the discriminant function can be used to represent all WOSF with the same output but different weight vectors and threshold values.

Table 1. The output and OS filters representation

Input $\vec{x}_i$	**Output**				
(x_1, x_2, x_3)	f_0	f_1	f_2	f_3	f_4
(0, 0, 0)	1	0	0	0	0
(0, 0, 1)	1	1	0	0	0
(0, 1, 0)	1	1	0	0	0
(0, 1, 1)	1	1	1	0	0
(1, 0, 0)	1	1	0	0	0
(1, 0, 1)	1	1	1	0	0
(1, 1, 0)	1	1	1	0	0
(1, 1, 1)	1	1	1	1	0

Table 2. The output and WOS filters representation with $\Omega = (1,3,1)$

Input $\vec{x}_i$	**Output**						
(x_1, x_2, x_3)	$f_{\bar{\Omega},0}$	$f_{\bar{\Omega},1}$	$f_{\bar{\Omega},2}$	$f_{\bar{\Omega},3}$	$f_{\bar{\Omega},4}$	$f_{\bar{\Omega},5}$	$f_{\bar{\Omega},6}$
(0, 0, 0)	1	0	0	0	0	0	0
(0, 0, 1)	1	1	0	0	0	0	0
(0, 1, 0)	1	1	1	1	0	0	0
(0, 1, 1)	1	1	1	1	1	0	0
(1, 0, 0)	1	1	0	0	0	0	0
(1, 0, 1)	1	1	1	0	0	0	0
(1, 1, 0)	1	1	1	1	1	0	0
(1, 1, 1)	1	1	1	1	1	1	0

4 An Illustrative Example

From table 2, one takes a WOSF $f_{\Omega,4}(\vec{x}_i)$ that has a pack of outputs $(0\ 0\ 0\ 1\ 0\ 0\ 1\ 1)$. Composing of the inputs $\vec{x}_i \in \{0,\ 1\}^3$ and outputs to form a training set $S = \{(\vec{x}_i, y_i)\}_{i=1}^{8}$, for SVM training, one generates a hyperplane with normal vector $\vec{w}* = (1\ 2\ 1)$ and bias $b* = -1$. Hence the discriminant function can be written as

$$f*_{(121),-1}(\vec{x}_i) = <(1\ 2\ 1),(x_1x_2x_3)> + (-1)$$

Because a filter is characterized by the outputs $(0\ 0\ 0\ 1\ 0\ 0\ 1\ 1)$, the sign of discriminant function $\text{sgn}(f*_{(121),-1}(\vec{x}_i))$ can represent all WOS filters with the same outputs but different weight vectors Ω. For example, the WOSF $f_{(121),3}(\vec{x}_i)$, and $f_{(131),4}(\vec{x}_i)$ generate the same outputs both of which can be represented by the sign of discriminant function $\text{sgn}(f*_{(121),-1}(\vec{x}_i))$. The WOSF of traditional sense and SVM sense are listed in table 3.

Table 3. The WOSF representation of SVM sense and traditional sense with the same output

	Output				
	(01111111)	(00110111)	(00110011)	(00010011)	(00000001)
	SVM sense WOSF $\text{sgn}(f_{\vec{w}*,b*}(\vec{x}_i))$				
$\vec{w}*$	(111)	(121)	(010)	(121)	(111)
$b*$	2	1	0	−1	−2
Filter	$\text{sgn}(f_{(111),2})$	$\text{sgn}(f_{(121),1})$	$\text{sgn}(f_{(010),0})$	$\text{sgn}(f_{(121),-1})$	$\text{sgn}(f_{(111),-2})$
	Traditional sense WOSF $f_{\Omega,r'}(\vec{x}_i)$				
Ω	(131)	(131)	(131)	(131)	(131)
r'	1	2	3	4	5
Filter	$f_{(131),1}$	$f_{(131),2}$	$f_{(131),3}$	$f_{(131),4}$	$f_{(131),5}$
Ω	(324)	(231)	(120)	(121)	(111)
r'	2	3	2	3	3
Filter	$f_{(324),2}$	$f_{(231),3}$	$f_{(120),2}$	$f_{(121),3}$	$f_{(111),3}$
Ω	(543)	(243)	(241)	(243)	(213)
r'	3	4	4	6	6
Filter	$f_{(543),3}$	$f_{(243),4}$	$f_{(241),4}$	$f_{(243),6}$	$f_{(213),6}$
Ω	(474)	(154)	(252)	(252)	(121)
r'	4	5	5	7	4
Filter	$f_{(474),4}$	$f_{(154),5}$	$f_{(252),4}$	$f_{(252),7}$	$f_{(121),4}$
Ω	(589)	(364)	(263)	(465)	(225)
r'	5	6	6	10	9
Filter	$f_{(589),5}$	$f_{(364),6}$	$f_{(263),6}$	$f_{(465),10}$	$f_{(225),9}$

In table 3, the element “0” replaces the element “-1” to express the context. For the pack of outputs $(0\ 0\ 0\ 1\ 0\ 0\ 1\ 1)$, the represented WOSF merely uses a normal vector $\vec{w}^* = (1\ 2\ 1)$ and bias $b^* = -1$ in SVM sense. Compared with the traditional representation, the WOS filters need lots of different weight vectors Ω and threshold values r', as $\Omega = (1\ 3\ 1)$, $r' = 4$ or $\Omega = (2\ 4\ 3)$, $r' = 6$ or $\Omega = (2\ 5\ 2)$, $r' = 7$ ….. and so forth. Obviously, only few parameters are able to represent all WOSF having the same outputs but different weight vectors. Other WOSF corresponding different outputs also are listed in table 3. In table 3, each column contains a pack of outputs, a corresponding WOSF in SVM sense, and some WOSF in traditional sense.

5 Conclusion

In traditional sense, each weight vector and threshold value can represent a WOSF. Various weight vectors and threshold values can represent different WOSF, but the outputs are the same. To reduce the numbers of weight vectors and threshold values and use other form to represent WOSF that is the main purpose of this research.

This paper proposes a representation based on the properties of maximal margin classification of SVM. Through a truth table generated by a linearly separable Boolean function, one takes each input data and related outputs to form a training set for SVM training. One thus generates an optimal normal vector and an optimal bias. The normal vector and bias can represent all WOS filters with same outputs but different weight vectors and threshold values. Compared to the known representation on [5]-[8], the proposed method can characterize WOSF with fewer parameters.

Reference

1. P. D. Wentd, E. J. Coyle, and N. C. Gallagher Jr. : Stack Filters. IEEE Trans. Acoust. Speech and Signal Processing, vol. 34. no. 4. (1986) 898-911
2. O. Yli-Harja, J. T. Astola, and Y. Neuvo.: Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation. IEEE Transactions on Signal Processing, vol. 39. (1991) 395-410
3. J. Nieweglowski, M. Gabbouj, and Y. Neuvo.: Weighted medias-positive Boolean functions conversion algorithms. Signal Processing, vol. 34. (1982) 149-161
4. P. D. Wentd.: Nonrecursive and recursive stack filters and their filtering behavior. IEEE Trans. Acoust. Speech and Signal Processing, vol. 38. no. 12, (1990) 2099-2107
5. P. Koivisto, and H. Huttunen: Design Of Weighted Order Statistic Filters By Training-Based Optimization. International Symposium on Signal Processing and its Application (ISSPA), Kuala Lumpur, Malaysia, (2001)
6. Michael Ropert, Francois Moreau de Saint-Martin, and Danielle Pele. A new representation of weighted order statistic filters. Signal Processing, vol. 54. (1996) 201-206
7. C. E. Savin, M. O. Ahmad, and M. N. S. Swamy. Design of Weighted Order Statistic Filters Using Linearly Separable Stack-Like Architecture. Circuits and Systems, Proceedings of the 37th Midwest Symposium, vol. 2. (1994) 753–756

8. Nina S. T. Hirata and Roberto Hirata Jr.: Design of Order Statistic Filters from Examples. Proceedings of the XVI Brazilian Symposium on Computer Graphics and Image Processing (*SIBGRAPI'03*), Brazilian, (2003) 24-26
9. Yih-Lon Lin, Wei-Chin Teng, Jyh-Horng Jeng, and Jer-Guang Hsieh.: Characterization of Canonical Robust Template Values for a Class of Uncoupled CNNs Implementing Linearly Separable Boolean Functions. WSEAS Transactions on Information Science and Applications, vol.7, no.2. (2005) 940-944

Detection Method and Removing Filter for Corner Outliers in Highly Compressed Video

Jongho Kim, Kicheol Jeon, and Jechang Jeong

Dept. Electrical and Computer Engineering, Hanyang University,
17 Haengdang, Seongdong, Seoul 133-791, Korea
{angel, hobohemia, jjeong}@ece.hanyang.ac.kr

Abstract. We propose a detection method and a removal filter for corner outliers in order to improve visual quality in highly compressed video. Corner outliers are detected using the direction of edge going through a block-corner and the properties of blocks around the block-corner. The proposed filter for removing corner outliers, which compensates the stair-shaped discontinuities around edges using the adjacent pixels, is applied to the detected area. Simulation results show that the proposed method improves, particularly in combination with deblocking filters, the visual quality remarkably.

Keywords: corner outlier, low bit-rate video, block-based coding, MPEG-4 video.

1 Introduction

Most video coding standards, which have a hybrid structure, adopt block-based motion compensated prediction and transform. Consequently, the block-based processing generates undesired artifacts such as blocking artifacts, ringing noise, and corner outliers, particularly in very low bit-rate video. Blocking artifacts are grid noise along block boundaries in relatively flat areas; ringing noise is the Gibb's phenomenon due to truncation of high-frequency coefficients by quantization; corner outlier is a special case of blocking artifacts at the cross-point of a block-corner and a diagonal edge. To reduce the blocking artifacts and the ringing noise, a number of studies have been carried out in spatial domain [1, 2, 5] and transform domain [3, 4], respectively. However, the corner outliers are still visible in some video sequences, since a deblocking filter is not applied to the areas including a large difference at a block boundary in order to avoid undesired blurring [1]. Hardly any studies have been carried out on removing the corner outliers although the artifacts degrade visual quality considerably because the corner outliers appear just in limited areas and the peak signal-to-noise ratio (PSNR) improvement is somewhat small. Therefore, we propose an effective corner outlier removal filter with a simple detection method to improve visual quality in very low bit-rate video. The proposed method can be used along with various deblocking [1-4] and deringing filters [5] for more improvement of visual quality.

L.-W. Chang, W.-N. Lie, R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1177 – 1184, 2006.

The remaining parts of the paper are as follows. We present a detection method for corner outliers based on the pre-defined patterns, i.e., the direction of an edge going through a block-corner, in Section 2. Section 3 describes, in detail, the proposed filter to remove the corner outliers. Simulation results and conclusions are given in Section 4 and Section 5, respectively.

2 Definition and Observations on the Corner Outliers

Consider the original and its reconstructed frames illustrated in Fig. 1, where a diagonal edge goes through a block-corner. The edge occupies large areas in blocks **B** and **C**, whereas it occupies very small areas (d_0 in Fig. 1) in block **D**. If block **D** is flat except d_0, the AC coefficients of DCT of block **D** are mainly related to d_0, and their values are small. Since most small AC coefficients are truncated by quantization in very low bit-rate coding, the area of d_0, which represents the edge in block **D**, cannot be reconstructed as shown in Fig. 1(b). As a result, the visually annoying stair-shaped artifact is produced around the block-corner. Such artifacts are called corner outliers.

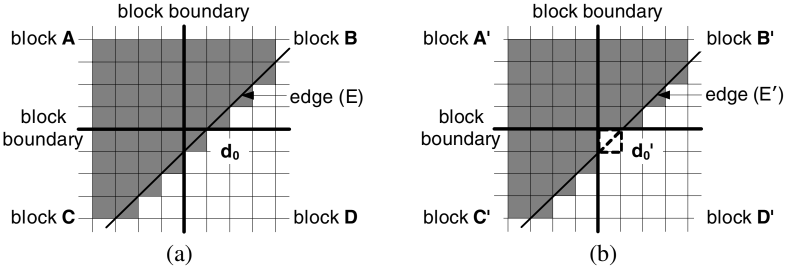
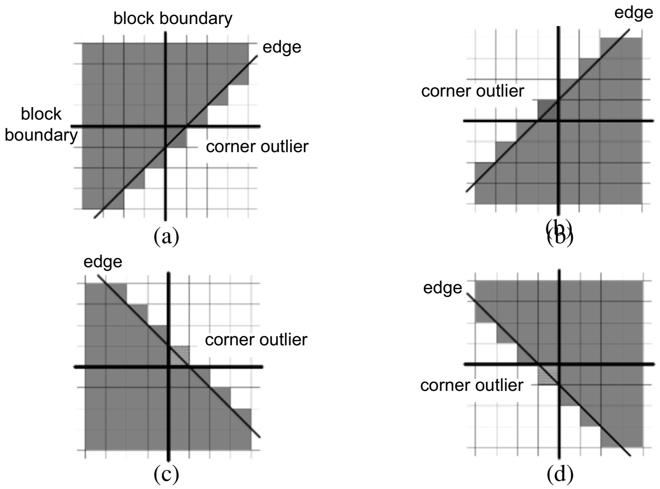


Fig. 1. Conceptual illustration of a corner outlier, (a) an edge going through a block-corner in an original frame, (b) an corner outlier located in d_0' in a reconstructed frame

To detect and remove the corner outliers, we find two major observations on the artifacts as follows. First, there is a large difference between boundary values of the block including the corner outlier and the other three blocks around the cross-point. For example, the difference between d_0' and its upper pixel or between d_0' and its left pixel is large as shown in Fig. 1(b). Second, corner outliers are more noticeable in flat areas, that is, each block around the cross-point is relatively flat. We examine the properties of the blocks in terms of the observations around every cross-point to detect the corner outliers appropriately. In addition, since corner outliers are prominent when a diagonal edge occupies block areas unequally around a cross-point, we deal with four types, as depicted in Fig. 2, based on the edge direction. We detect the actual corner outlier, which satisfies the proposed detection conditions among each detection type. Dealing with the pre-defined detection types instead of detecting an edge precisely has an advantage in reduction of computational complexity.



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

Fig. 2. Detection types based on the edge direction, (a) Lower 45° direction, (b) Upper 45° direction, (c) Upper 135° direction, (d) Lower 135° direction

3 Detection Method and Removal Filter for Corner Outliers

3.1 Detection Method for Corner Outliers

To detect corner outliers based on the first observation, we obtain the average values of the four pixels around the cross-point, which are represented as the shaded areas in Fig. 3, by

$$A_{avg} = \frac{1}{4}\sum_{i=1}^{4} a_i\,,\quad B_{avg} = \frac{1}{4}\sum_{i=1}^{4} b_i\,,\quad C_{avg} = \frac{1}{4}\sum_{i=1}^{4} c_i\,,\quad D_{avg} = \frac{1}{4}\sum_{i=1}^{4} d_i \tag{1}$$

where each capital and small letter denotes blocks and pixels, respectively, that is, A_{avg} is an average from a_1 to a_4 of block **A**, B_{avg} is an average from b_1 to b_4 of block **B**, and so on. Then, to determine the detection type among the four cases, we examine the differences between the average values of the corner-outlier candidate block and its neighboring blocks, using the equations listed in Table 1. By using the average values around the cross-point instead of each pixel value, we can reduce detection errors in complex areas.

In Table 1, QP denotes the quantization parameter. For example, when block A has a corner outlier, the differences between A_{avg} and B_{avg} and between A_{avg} and C_{avg} is large by the first observation, thereby we consider the block **A** has a corner outlier. Practically, since we do not know the corner-outlier candidate block, the four cases listed in Table 1 should be investigated. If two corner outliers appear at one cross-point, the artifacts are arranged on diagonally opposite sides because the corner outliers cannot be placed vertically or horizontally. In this case, the proposed method can detect both artifacts without additional computations, since we examine the four cases independently using the equations in Table 1.

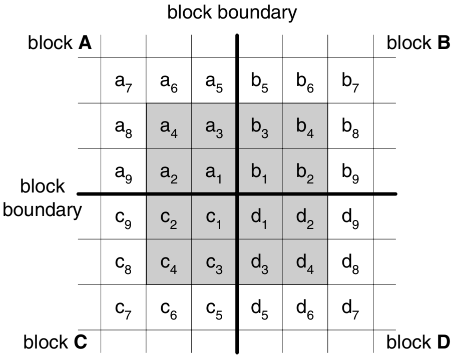


Fig. 3. Pixel arrangement for detecting and filtering corner outliers

Table 1. Detection criterion according to the detection type

Block containing the artifact	Detection type	Detection criterion
A	Upper 45°	$\lvert A_{avg} - B_{avg}\rvert > 2\mathrm{QP}$ and $\lvert A_{avg} - C_{avg}\rvert > 2\mathrm{QP}$
B	Upper 135°	$\lvert B_{avg} - A_{avg}\rvert > 2\mathrm{QP}$ and $\lvert B_{avg} - D_{avg}\rvert > 2\mathrm{QP}$
C	Lower 135°	$\lvert C_{avg} - A_{avg}\rvert > 2\mathrm{QP}$ and $\lvert C_{avg} - D_{avg}\rvert > 2\mathrm{QP}$
D	Lower 45°	$\lvert D_{avg} - B_{avg}\rvert > 2\mathrm{QP}$ and $\lvert D_{avg} - C_{avg}\rvert > 2\mathrm{QP}$

To detect corner outliers based on the second observation, we examine whether the block satisfying the condition in Table 1 is flat or not. That is, when the candidate block is **A**, we examine flatness of block **A** with respect to the pixel including the corner outlier using

$$A_{flat} = \sum_{i=2}^{9} \left| a_1 - a_i \right| \tag{2-1}$$

where a_1 is the pixel including the corner outlier. We regard the block as flat when A_{flat} is smaller than QP. In case where another block is determined as a candidate block, we examine whether each of the followings is smaller than QP or not.

$$B_{flat} = \sum_{i=2}^{9} \left| b_1 - b_i \right|, \quad C_{flat} = \sum_{i=2}^{9} \left| c_1 - c_i \right|, \quad D_{flat} = \sum_{i=2}^{9} \left| d_1 - d_i \right| \tag{2-2}$$

where each capital and small letter denotes blocks and pixels, respectively, and QP is the quantization parameter. When the condition of Table 1 is satisfied and the value of (2) is smaller than QP, we regard the block as including a corner outlier, and then the proposed filter is applied to the block in order to remove the artifact.

3.2 The Proposed Corner Outlier Removal Filter

To remove the corner outliers, we propose a filter that updates pixels of the stair-shaped discontinuity using neighboring pixels. To reduce computational complexity, we apply the proposed filter under the assumption that the edge has a diagonal direction instead of detecting the actual edge direction of neighboring blocks. This is reasonable because:

1. Corner outliers created by a diagonal edge are more noticeable than the artifacts created by a horizontal or vertical edge at a cross-point.
2. Various deblocking filters can remove corner outliers created by a horizontal or vertical edge.

According to the above assumptions and the detection results obtained in Section 2, when block **A** includes a corner outlier as shown in Fig. 2(b), the pixels in block **A** are replaced by

$$\begin{cases} a_1' = (b_1 + b_2 + c_1 + d_1 + d_2 + c_3 + d_3 + d_4)/8 \\ a_2' = (a_2 + a_1' + b_1 + c_2 + c_1 + d_1 + c_4 + c_3 + d_3)/9 \\ a_3' = (a_3 + b_3 + b_4 + a_1' + b_1 + b_2 + c_1 + d_1 + d_2)/9 \\ a_4' = (a_4 + a_3' + b_3 + a_2' + a_1' + b_1 + c_2 + c_1 + d_1)/9 \\ a_5' = (a_5 + b_5 + b_6 + a_3' + b_3 + b_4 + a_1' + b_1 + b_2)/9 \\ a_9' = (a_9 + a_2' + a_1' + c_9 + c_2 + c_1 + c_8 + c_4 + c_3)/9 \end{cases} \tag{3}$$

where each index follows that of Fig. 3. For blocks **B**, **C**, and **D**, the filtering methods are similar to (3) and actual equations are as follows: in case where block **B** includes a corner outlier, as seen in Fig. 2(c), the pixels in block **B** are replaced by

$$\begin{cases} b_1' = (a_1 + a_2 + d_1 + c_1 + c_2 + d_3 + c_3 + c_4)/8 \\ b_2' = (b_2 + b_1' + a_1 + d_2 + d_1 + c_1 + d_4 + d_3 + c_3)/9 \\ b_3' = (b_3 + a_3 + a_4 + b_1' + a_1 + a_2 + d_1 + c_1 + c_2)/9 \\ b_4' = (b_4 + b_3' + a_3 + b_2' + b_1' + a_1 + d_2 + d_1 + c_1)/9 \\ b_5' = (b_5 + a_5 + a_6 + b_3' + a_3 + a_4 + b_1' + a_1 + a_2)/9 \\ b_9' = (b_9 + b_2' + b_1' + d_9 + d_2 + d_1 + d_8 + d_4 + d_3)/9 \end{cases} \tag{4}$$

in case where block **C** includes a corner outlier, as seen in Fig. 2(d), the pixels in block **C** are replaced by

$$\begin{cases} c_1' = (d_1 + d_2 + a_1 + b_1 + b_2 + a_3 + b_3 + b_4)/8 \\ c_2' = (c_2 + c_1' + d_1 + a_2 + a_1 + b_1 + a_4 + a_3 + b_3)/9 \\ c_3' = (c_3 + d_3 + d_4 + c_1' + d_1 + d_2 + a_1 + b_1 + b_2)/9 \\ c_4' = (c_4 + c_3' + d_3 + c_2' + c_1' + d_1 + a_2 + a_1 + b_1)/9 \\ c_5' = (c_5 + d_5 + d_6 + c_3' + d_3 + d_4 + c_1' + d_1 + d_2)/9 \\ c_9' = (c_9 + c_2' + c_1' + a_9 + a_2 + a_1 + a_8 + a_4 + a_3)/9 \end{cases} \tag{5}$$

and in case where block **D** includes a corner outlier, as seen in Fig. 2(a), the pixels in block **D** are replaced by

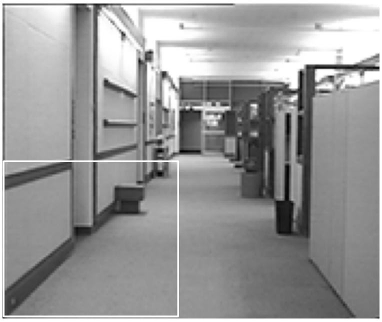
$$\begin{cases} d_1' = (c_1 + c_2 + b_1 + a_1 + a_2 + b_3 + a_3 + a_4)/8 \\ d_2' = (d_2 + d_1' + c_1 + b_2 + b_1 + a_1 + b_4 + b_3 + a_3)/9 \\ d_3' = (d_3 + c_3 + c_4 + d_1' + c_1 + c_2 + b_1 + a_1 + a_2)/9 \\ d_4' = (d_4 + d_3' + c_3 + d_2' + d_1' + c_1 + b_2 + b_1 + a_1)/9 \\ d_5' = (d_5 + c_5 + c_6 + d_3' + c_3 + c_4 + d_1' + c_1 + c_2)/9 \\ d_9' = (d_9 + d_2' + d_1' + b_9 + b_2 + b_1 + b_8 + b_4 + b_3)/9 \end{cases} \tag{6}$$

In each equation, the indices follow those of Fig. 3.

4 Simulation Results

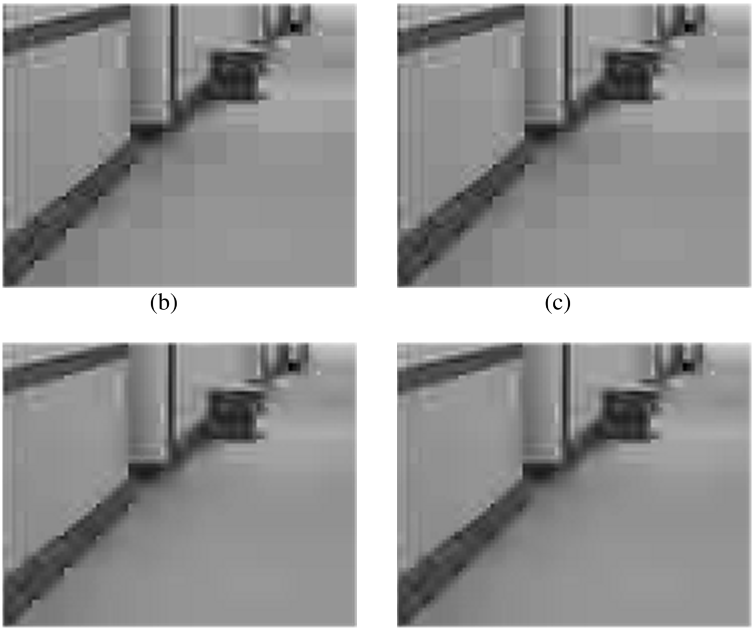
The proposed method was applied to ITU test sequences at various bit-rates. To evaluate the proposed method, each test sequence was coded using the MPEG-4 verification model (VM) [6] with two coding modes: IPPP…, i.e., all frames are inter-frame coded except the first frame, and I-only, i.e., all frames are intra-frame coded. In each mode, we applied the proposed filter to the reconstructed frames with none of the coding options being switched on and with the deblocking filter [2] being switched on, respectively. To arrive at a certain bit-rate, an appropriate quantization parameter was chosen and kept constant throughout the sequence. This can avoid possible side effects from typical rate control methods.

The simulation results for IPPP… coded sequences are summarized in Table 2. It can be seen that PSNR results for the luminance component are increased by up to 0.02dB throughout the sequences. Just a slight improvement in PSNR is obtained due to limitation of the area satisfying the filtering conditions. However, the proposed filter considerably improves subjective visual quality, particularly for sequences with apparent diagonal edges such as *Hall Monitor*, *Mother & Daughter*, *Foreman*, etc. For emphasizing the effect of the proposed filter, partially enlarged frames of the first frame of *Hall Monitor* sequence under each method, i.e., the result of the proposed method without and with deblocking filter, respectively, are shown in Fig. 4.



(a)

Fig. 4. Result images for *Hall Monitor* sequence (a) Original sequence (QCIF, QP=17), (b) and (d) partially enlarged images of MPEG-4 reconstructed images without and with the MPEG-4 deblocking filter, respectively, (c) and (e) partially enlarged images of the proposed method without and with the MPEG-4 deblocking filter, respectively



(b)

(c)

(d)

(e)

Fig. 4. (*continued*)

Table 2. PSNR results for IPPP… case

Sequence	QP	Bit-rate (kbps)	PSNR_Y (dB)			
			No filtering	No filtering + proposed filter	Deblocking	Deblocking + proposed filter
Hall Monitor	18	9.39	29.8728	29.8851	30.1957	30.2090
Mother & Daughter	16	9.45	32.2256	32.2469	32.3684	32.3894
Container Ship	17	9.8	29.5072	29.5082	29.7281	29.7291
Hall Monitor	9	24.29	34.0276	34.0325	34.1726	34.1835
Mother & Daughter	8	23.83	35.3303	35.3316	35.2686	35.2708
Container Ship	10	21.61	32.5701	32.5711	32.6003	32.6013
Foreman	14	46.44	30.6237	30.6252	31.0727	31.0739
Coastguard	14	44.68	29.0698	29.0763	29.1562	29.1586
Hall Monitor	12	47.82	33.8216	33.8236	34.0921	34.0948
News	19	47.21	31.2192	31.2202	31.3516	31.3528
Foreman	12	64.65	31.4786	31.4798	31.7587	31.7598
News	16	63.14	32.0648	32.0658	32.1233	32.1243

Table 3 shows the results for I-only coded sequences of *Hall Monitor* and *Foreman*. The results are similar to the IPPP… coded case. The proposed filtering conditions are satisfied at low QP for the sequences that include low spatial details such as *Hall Monitor*, since each frame of the sequences is relatively flat originally.

On the other hand, the proposed filtering conditions are satisfied at relatively high QP for the sequences that include medium or high spatial details such as *Foreman*, since each frame of the sequences is flattened at high QP. This tendency maintains in the sequences with or without the deblocking filter.

Table 3. PSNR results for I-only case

Sequence	Method	PSNR_Y (dB)			
		QP=12	QP=17	QP=22	QP=27
Hall Monitor	No filtering	32.8170	30.5284	28.9326	27.6323
	No+proposed	32.8396	30.5381	28.9374	27.6344
	Deblocking	33.2223	30.9624	29.4025	28.1188
	Deblocking+proposed	33.2422	30.9848	29.4052	28.1230
Foreman	No filtering	32.1830	30.0531	28.6406	27.5515
	No+proposed	32.1862	30.0606	28.6540	27.5684
	Deblocking	32.5741	30.5267	29.1635	28.1405
	Deblocking+proposed	32.5760	30.5316	29.1745	28.1526

5 Conclusions

The corner outliers are very annoying visually in highly compressed video although they appear in limited areas. To remove the corner outliers, we have proposed a simple and effective post-processing method, which includes a detection method and a compensation filter. The proposed method, particularly in combination with the deblocking filter, further improves both objective and subjective visual quality.

Acknowledgments. This work was supported in part by the Human Resource Development Project for IT SoC Key Architect under Korea IT Industry Promotion Agency.

References

1. Park, H., Lee, Y.: A Postprocessing Method for Reducing Quantization Effects in Low Bit-rate Moving Picture Coding. IEEE Trans. Circuits and Syst. Video Technol., Vol. 9. (1999) 161-171
2. Kim, S., Yi, J., Kim, H., Ra, J.: A Deblocking Filter with Two Separate Modes in Block-based Video Coding. IEEE Trans. Circuits and Syst. Video Technol., Vol. 9. (1999) 156-160
3. Zhao, Y., Cheng, G., Yu, S.: Postprocessing Technique for Blocking Artifacts Reduction in DCT Domain. IEE Electron. Lett., Vol. 40. (2004) 1175-1176
4. Wang, C., Zhang, W.-J., Fang, X.-Z.: Adaptive Reduction of Blocking Artifacts in DCT Domain for Highly Compressed Images. IEEE Trans. Consum. Electron., Vol. 50. (2004) 647-654
5. Kaup, A.: Reduction of Ringing Noise in Transform Image Coding Using Simple Adaptive Filter. IEE Electron. Lett., Vol. 34. (1998) 2110-2112
6. Text of MPEG-4 Video Verification Model ver.18.0. ISO/IEC Doc. N3908, (2001)

Accurate Foreground Extraction Using Graph Cut with Trimap Estimation

Jung-Ho Ahn and Hyeran Byun

Dept. of Computer Science, Yonsei University, Seoul, Korea, 126-749

Abstract. This paper describes an accurate human silhouette extraction method as applied to video sequences. In computer vision applications that use a static camera, the background subtraction method is one of the most effective ways of extracting human silhouettes. However it is prone to errors so performance of silhouette-based gait and gesture recognition often decreases significantly. In this paper we propose two-step segmentation method: trimap estimation and fine segmentation using a graph cut. We first estimated foreground, background and unknown regions with an acceptable level of confidence. Then, the energy function was identified by focussing on the unknown region, and it was minimized via the graph cut method to achieve optimal segmentation. The proposed algorithm was evaluated with respect to ground truth data and it was shown to produce high quality human silhouettes.

1 Introduction

Background subtraction has been widely used to detect and track moving objects obtained from a static camera. Recently background subtraction methods have been used to assist in human behavior analysis such as surveillance and gait and gesture recognition[10, 13, 18]. In gait and gesture recognition silhouettes are used to determine configurations of the human body. However these human silhouettes are often not accurate enough and recognition performance can decrease significantly when using these silhouettes [11]. Especially, shadows can not always be properly removed, and some parts of the silhouette can be lost when occluding object parts have similar colors as the occluded background areas. Figure 1 demonstrates these problems.

Background subtraction has long been an active area of research. Horprasert *et al.* [6] proposed a robust background subtraction and shadow detection method which was applied to an object tracking system together with an appearance model [13]. The adaptive background subtraction method using the Gaussian Mixture Model (GMM) [14] was presented and it was applied to foreground analysis with intensity and texture information [16]. A joint color-with-depth observation space [7] and an efficient nonparametric background color model [5] were also proposed to extract foreground objects.

In this paper we propose a foreground segmentation that consists of two steps; trimap estimation and fine segmentation using a graph cut. In trimap estimation we estimate the confident foreground and background regions and leave the

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1185–1194, 2006.

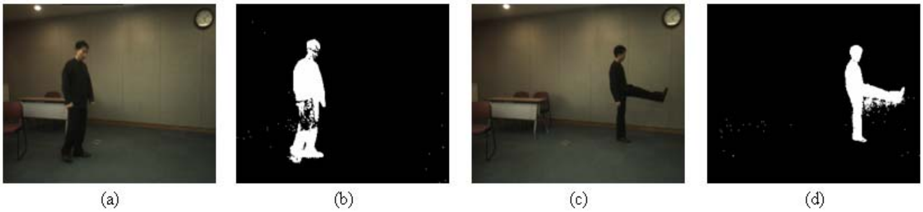


Fig. 1. Problems of the background subtraction method to extract human silhouettes. In (a) and (b) some parts of the human silhouette disappear when the color of these parts is too similar to that of the occluded background area. (c) and (d) shows the shadow problem. These silhouettes were obtained with the Horprasert's algorithm [6].

dubious area unknown. The amount of the estimated foreground and background regions is related to the level of confidence. Fine segmentation focus on the unknown regions and uses the energy minimization technique. Throughout this paper it is assumed that a static background is available, images are captured by a static mono camera, and only one person enters a scene.

As a pioneer work of object segmentation using graph cut, the user-interactive segmentation technique was proposed by Boykov and Jolly [1]. GrabCut [12] and lazy snapping[9] were other user-interactive image cutout systems based on a graph cut. Lazy snapping used a graph cut formulation based on regions generated by the watershed algorithm, instead of the image pixels. In this paper we also perform image segmentation but it is for estimating the trimap. Recently, the Layered Graph Cut(LGC) method based on color, contrast and stereo matching [8] information has been proposed to infer the foreground by using a pair of fixed webcams.

Section 2 describes the trimap concept and the proposed graph cut framework using the trimap information. Section 3 explains the concrete trimap estimation method using region likelihood. Experimental results are given in section 4and then conclusions are presented in section 5.

2 Graph Cut Segmentation with Estimated Trimap

2.1 Estimated Trimap

In natural image matting[4][15] with still images the trimap is supplied by the user. This trimap partitions the image into three regions: firm foreground, firm background and unknown regions. In unknown regions, the matte can be estimated using the color statistics in the known foreground and background regions. In this paper we attempt to estimate the trimap in video applications without user interaction.

Foreground segmentation refers to a binary labeling(classification) process that assigns all the pixels in a given image to either foreground or background. It is very hard to label *all* the pixels in the image correctly, but *some* parts of the

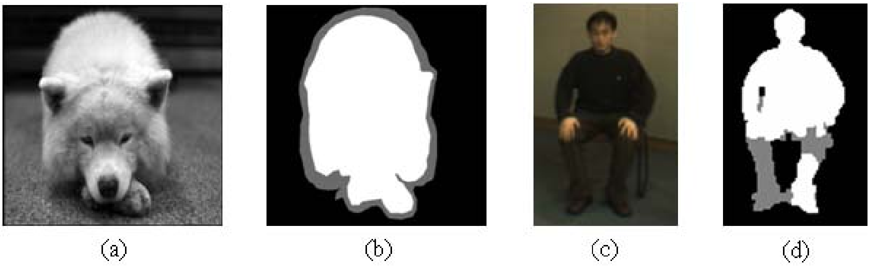


Fig. 2. Trimap Comparison. (a) and (b) show typical trimaps in natural image matting [15], whereas (c) and (d) shows the trimaps used in the proposed algorithm.

image can be easily labeled with simple ideas or features. In trimap estimation we pre-determine the labels of each of the pixels in the area that can be easily labeled. Then the trimap information is reflected into the energy function for fine segmentation. In section 3 we propose a trimap estimation method that uses the background model. Traditionally the unknown regions of the trimap is located between foreground and background areas but they can theoretically be placed anywhere in our estimated trimap. Figure 2 shows some typical examples of the trimaps of image matting and the proposed method.

The trimap $\mathcal{T}$ can be viewed as the function from the set of pixels $\mathcal{P}$ of the image to be segmented to the label set $\mathcal{L}_T$

$$\mathcal{T} : \mathcal{P} \rightarrow \mathcal{L}_T = \{-1, 0, 1\} \tag{1}$$

where -1, 0, and 1 represent unknown, background and foreground respectively. In every frame we estimate the trimap $\mathcal{T}$ and define the sets $\mathcal{O}$ and $\mathcal{B}$ by

$$\mathcal{O} = \{p \in \mathcal{P} | \mathcal{T}(p) = 1\}, \qquad \mathcal{B} = \{p \in \mathcal{P} | \mathcal{T}(p) = 0\}. \tag{2}$$

The set of unknown pixels $\mathcal{U}$ is defined by $\mathcal{P} - (\mathcal{O} \cup \mathcal{B})$. We call the pixels in $\mathcal{O}$ the foreground seeds. The pixels in $\mathcal{B}$ are called the background seeds.

2.2 Graph Cut with the Estimated Trimap

In this section we describe the energy function for fine segmentation. The energy function is similar to that used in the GrabCut method [12] but it explores both trimap information and the color likelihoods. We consider a standard neighborhood system $\mathcal{N}$ of all unordered pairs $\{p, q\}$ of neighboring pixels, and define $\mathbf{z} = (\mathbf{z}_1, \cdots, \mathbf{z}_{|\mathcal{P}|})$ as the image where $\mathbf{z}_n$ is the RGB color vector for the nth pixel and $f = (f_1, f_2, \cdots, f_{|\mathcal{P}|})$ as a binary vector whose components f_p specify label assignments to pixels p in $\mathcal{P}$. Each f_p can be either 1 or 0 where 1 represents the foreground and and 0 represents the background. The vector f defines a segmentation. Given an image, we seek the labeling f that minimizes the energy

$$E(f) = \gamma D(f) + V(f) \tag{3}$$

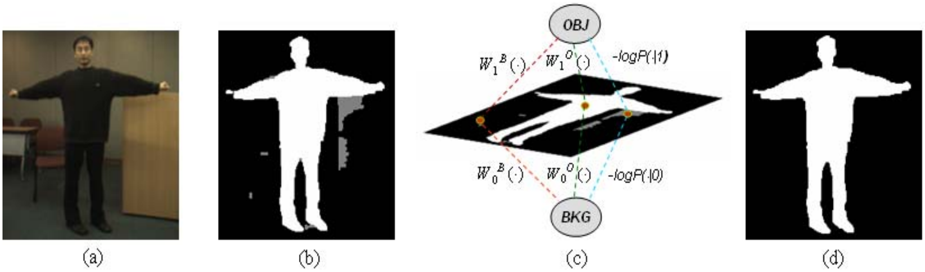


Fig. 3. Graph cut with the estimated trimap information. (a) input image, (b) trimap; white, black, gray areas indicate estimated foreground, background and unknown regions, respectively. (c) graph cut framework with the estimated trimap information, (d) the extracted foreground region.

where coefficient γ specifies the relative importance of the data term $D(f)$ and the smoothness term $V(f)$. The form of $D(f)$ and $V(f)$ are given by

$$D(f) = \sum_{p \in \mathcal{P}} D_p(f_p)$$

$$V(f) = \sum_{\{p,q\} \in \mathcal{N}} \delta(f_p, f_q) V_{p,q}(f_p, f_q).$$

where $\delta(f_p, f_q)$ denotes the delta function defined by 1 if $f_p \neq f_q$ and 0 otherwise. Given an image, it is necessary that the segmentation boundaries align with contours of high image contrast. This process is modeled using the smoothness term $V_{p,q}$ defined by

$$V_{p,q}(f_p, f_q) = \exp\left(-||\mathbf{z}_p - \mathbf{z}_q||^2/\beta\right) \tag{4}$$

where the constant β is chosen by the expectation of $2||\mathbf{z}_p - \mathbf{z}_q||^2$ over all $\{p, q\} \in \mathcal{N}$. The data term D_p measures how well label f_p fits pixel p given the observed data $\mathbf{z}_p$. We model the data term by using the given trimap $\mathcal{T}$ and the foreground and background color likelihoods of $P(\cdot|1)$ and $P(\cdot|0)$, respectively. The likelihoods are modeled by using Gaussian mixtures in the RGB color space, learned from image frames labelled from earlier in the sequence. Using the trimap information the data term D_p can be defined as

$$D_p(f_p) = \begin{cases} -\log P(\mathbf{z}_p|f_p) & \text{if } p \in \mathcal{U} \\ W_{f_p}^{\mathcal{O}}(\mathbf{z}_p|f_p) & \text{if } p \in \mathcal{O} \\ W_{f_p}^{\mathcal{B}}(\mathbf{z}_p|f_p) & \text{if } p \in \mathcal{B} \end{cases} \tag{5}$$

In the above equation $W_{f_p}^{\mathcal{O}}(\mathbf{z}_p)$'s and $W_{f_p}^{\mathcal{B}}(\mathbf{z}_p)$ are defined by

$$W_{f_p}^{\mathcal{O}}(\mathbf{z}_p) = -W_{f_p} \log P(\mathbf{z}_p|f_p)$$

$$W_{f_p}^{\mathcal{B}}(\mathbf{z}_p) = -W_{1-f_p} \log P(\mathbf{z}_p|f_p) \tag{6}$$

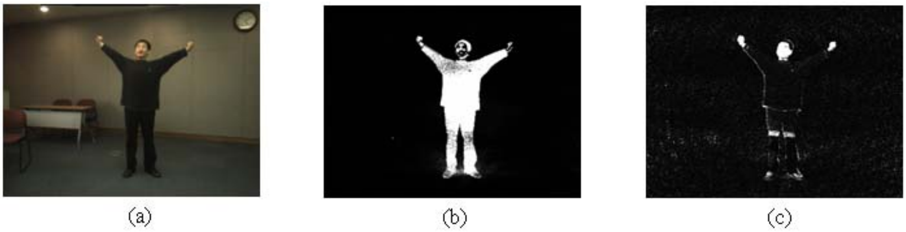


Fig. 4. Pixel foreground likelihoods: (a) image of the 250th frame in the test data $JH1$, (b) brightness distortion likelihood $-\log f^b(\alpha_p|p)$, (c) scaled chromaticity distortion likelihood $-\eta \log f^c(\gamma_p|p)$, where $\eta = 5$. The likelihoods are truncated by 255.

for all $p \in \mathcal{P}$ and $W_1 > 1 > W_0$. This data term model boosts the likelihood of $-\log P(\cdot|f_p)$ but suppresses $-\log P(\cdot|1 - f_p)$ for the pixel p of $\mathcal{T}(p) = f_p$ where f_p= 0 or 1. The estimated foreground and background seeds can be incorrectly labeled, thus the trimap information is used to distort the color likelihoods in the data term and the final segmentation label is given by a graph cut.

Minimization of the energy (3) is done by a standard min-cut/max-flow algorithm [2]. Figure 3 (c) illustrates the proposed graph cut framework using the estimated trimap information.

3 The Trimap Estimation Method Using Region Likelihood

To extract an accurate foreground silhouette, it is very important that the set $\mathcal{O}$ in the trimap $\mathcal{T}$ should not contain any background pixels but should contain as many foreground pixels as possible, and vice versa for the set $\mathcal{B}$. To accomplish this, we estimate the seeds of $\mathcal{T}$ in the region units, instead of in the pixel units. Region unit processing can have the a denoising effect which can allow for recovering clear outlines of the foreground objects. This processing was motivated from the perceptual grouping principles for object segmentation of still images[17]. We first propose the background color distribution model in section 3.1, and the region likelihood is determined in section 3.2. Finally, the region decision function for trimap estimation is proposed in section 3.3.

3.1 Brightness and Chromaticity Background Likelihood

Horprasert et. al[6] proposed the statistical background model that separated the brightness from the chromaticity component. They modeled a pixel p by 4-tuple $< \mu_p, \sigma_p, a_p, b_p >$ where μ_p was the expected color value, σ_p was the standard deviation of the RGB color value, a_p was the variation of the brightness distortion, and b_p was the variation of the chromaticity distortion of pixel p.

Using the background model, we propose the foreground likelihood $l(p)$ for each pixel p. The object likelihood is decomposed of the brightness and chromaticity likelihoods. From the definitions of the distortions , the distribution

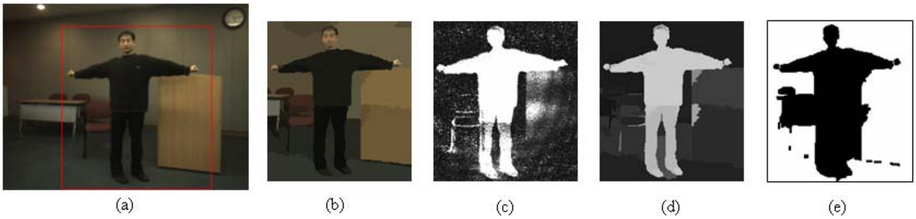


Fig. 5. Region unit processing for trimap estimation. (a) original image of the 581st frame in the test data *JH*3 with a bounding box, (b) mean shift segmentation of the image within the bounding box, (c) pixel foreground likelihoods $l(p)$'s, (d) naive region likelihood $L_p(R_i)$ (e) white area shows the regions that touch the bounding box.

f^b of the brightness distortion α of pixel p can be modelled using normal distribution of mean 1 and variance a_p^2, $\alpha \sim N(1, a_p^2)$. The distribution f^c of the chromaticity distortion γ at pixel p can be approximated by one-sided normal distribution of mean 0 and variance ${b_p}^2$,

$$f^c(\gamma|p) = \frac{2}{\sqrt{2\pi}b_p} \exp(-\gamma^2/(2{b_p}^2)), \qquad \gamma \geq 0. \tag{7}$$

Assuming that brightness distortion α_p and chromatic distortion γ_p are independent, the *naive* background probability density function f_B of pixel p can be given by

$$f_B(C_p|p) = f^b(\alpha_p|p) f^c(\gamma_p|p)\,, \tag{8}$$

where C_p is the RGB color vector of pixel p, and α_p and γ_p are calculated as in [6]. Figure 4 shows that the brightness and chromaticity distortions complement each other, thus our independence assumption can be empirically supported. The pixel foreground likelihood l of pixel p is given by

$$l(p) = -\log(f^b(\alpha_p|p)) - \eta \log(f^c(\gamma_p|p)), \tag{9}$$

where a constant η is introduced since the chromaticity distortion is relatively smaller than the brightness distortion in practice. Note that $l(p) = -\log(f_B(C_p|p))$ when $\eta = 1$.

3.2 Region Likelihoods

In every frame we perform image segmentation in order to partition each image into homogeneous small regions. We define $\mathcal{R} = \{R_i\}_{i \in I}$ as the set of the regions. For computational efficiency we perform it in the bounding box surrounding the foreground object. The bounding boxes are obtained by the maximal connected component of the foreground regions that are extracted by pixel-wise background subtraction. The foreground region likelihood $L(R_i)$ of region $R_i \in \mathcal{R}$ is calculated by

$$L(R_i) = \lambda_1 L_p(R_i) + \lambda_2 L_o(R_i). \tag{10}$$

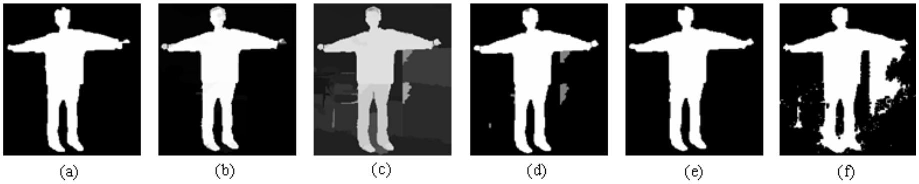


Fig. 6. Foreground segmentation with trimap estimation. (a) previous foreground object silhouette of the 580th frame, (b) regularization term $L_o(R_i)$ scaled by 255, (c) foreground region likelihood $L(R_i)$ truncated by 255. (d) the estimated trimap, (e) silhouette obtained by the proposed algorithm, (g) silhouette obtained by the Horprasert's algorithm[6]; deep shadow are on the floor and the teaching desk.

.

In the above equation, $L_p(R_i)$ is the *naive* foreground region likelihood given by the arithmetic mean of the pixel foreground likelihoods $l(p)$'s, i.e. $\sum_{p \in R_i} l(p)/n_{R_i}$, here n_{R_i} is the number of pixels in the region R_i. The naive region likelihood is not enough to decide the foreground object region especially when the occluding foreground parts are a similar color to the occluded background parts or when there are some deep shadows. Figure 5 shows an example of this. case. To overcome this problem we use the regularization term $L_o(R_i)$. This is the overlapping ratio of region R_i given by $n^o_{R_i}/n_{R_i}$, where $n^o_{R_i}$ is the number of pixels in region R_i that belong to the previous foreground object region. λ_2 is a regularization parameter. Figure 5 shows the pixel and naive foreground region likelihoods of an image. Some foreground regions with lower naive region likelihoods were supplemented by L_o in Fig. 6 (c). In the experiments we set $\lambda_1 = 0.8$ and $\lambda_2 = 30$.

3.3 Trimap Decision

By using the region likelihoods $L(R_i)$ in (10), the trimap can be estimated by using the region decision function $f_D : \mathcal{R} \rightarrow \{-1, 0, 1\}$ defined by,

$$f_D(R_i) = \begin{cases} 1 & \text{if } L(R_i) > T_U \\ 0 & \text{if } L(R_i) < T_L \\ -1 & \text{otherwise.} \end{cases} \tag{11}$$

where the thresholds T_U and T_L are related to the level of confidence on the trimap information. We set $T_U = 170$ and $T_L = 100$ in the experiments. Furthermore we define $\mathcal{R}_B$ as a collection of the regions that touch the bounding box. The regions in $\mathcal{R}_B$ are assumed to belong to the background area but when some region R_js belong to $\mathcal{R}_B$ and $f_D(R_i) = 1$ the regions are labeled as unknown. Figure 5 (e) shows the regions in $\mathcal{R}_B$ and an example of the estimated trimap is shown in Fig. 6 (d).

4 Experimental Results

Performance of the proposed method was evaluated with respect to the ground-truth segmentation of every tenth frame in each of five 700-frame test sequences

Table 1. Segmentation Error(%). The proposed algorithm(GT) is compared with the Horprasert's algorithm(HP) and the normal Graph Cut algorithm(GC) using color likelihoods on the test sequences of $JH1$, $JH2$, $JH3$, $GT1$ and $KC1$.

	$JH1$	$JH2$	$JH3$	$GT1$	$KC1$
GT	0.47	0.45	0.98	0.74	1.58
HP	2.59	5.22	1.67	1.66	3.01
GC	10.86	12.90	12.89	16.91	23.91

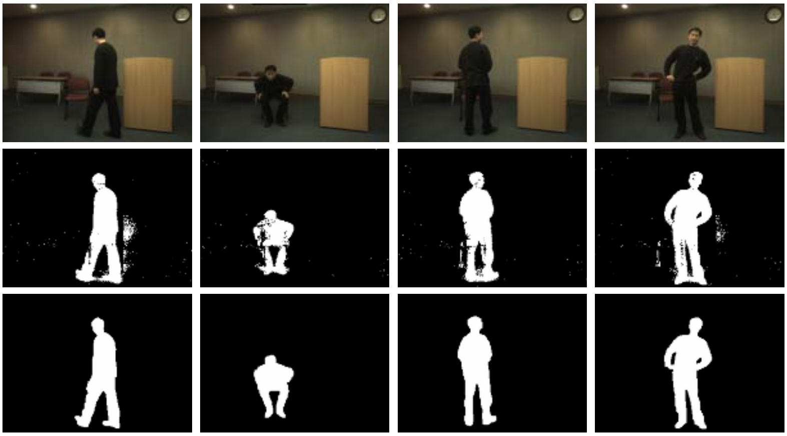


Fig. 7. Comparison of extracted human silhouettes. (Top) Four frames of the test sequence $JH3$. (Middle) Foreground extraction using HP[6]. (Bottom) Foreground extraction using the proposed algorithm.

of 320×240 images. The ground-truth data was labeled manually. Each pixel was labeled as foreground, background or unknown. The unknown label occurred in one plus pixel and one minus pixel along the ground-truth foreground object boundaries The five test sequences were labeled by $JH1$, $JH2$, $JH3$, $GT1$, $KC1$. Each was composed of different backgrounds and people. In all sequences, one person entered the scene, moved around and assumed natural poses, and the whole body was shown for gesture recognition. A person is shown under a light in *KC1* so that deep shadows are cast over the floor and wall.

Segmentation performance of the proposed method(GT) was compared with that of the Horprasert's background subtraction method(HP) [6] and the video version of the GrabCut method(GC) [12]. The only difference between GT and GC is that GT used a trimap but GC did not. Table 1 shows that the proposed method outperformed the both methods. The error rate was calculated within the bounding boxes only ignoring the mixed pixels. In the experiments we used ten Gaussian mixtures for color likelihood models and the mean shift segmentation method [3] was used to obtain the regions $\mathcal{R}^t$. Human silhouette results are

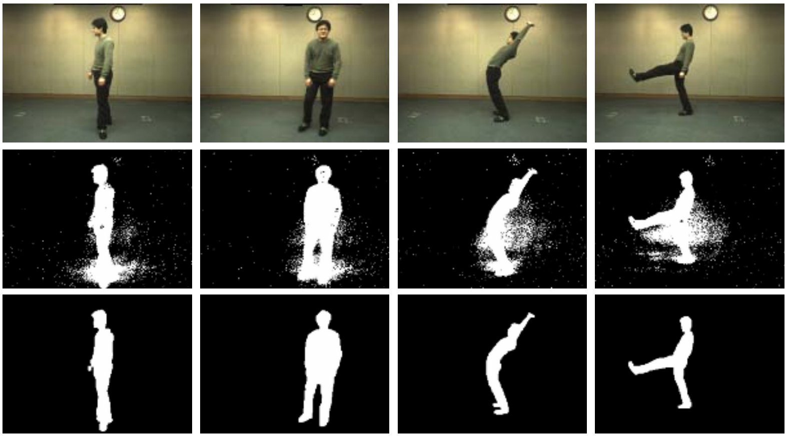


Fig. 8. Comparison of the silhouettes. (Top) Four frames of the test sequence $KC1$. (Middle) Foreground extraction using HP[6]. (Bottom) Foreground extraction using the proposed algorithm.

shown in Fig.7 and Fig. 8. The average running time of the proposed algorithm was 14.5 fps on a 2.8 GHz Pentium IV desktop machine with 1 GB RAM.

5 Conclusions

This paper has addressed accurate foreground extraction in video applications. We proposed a novel foreground segmentation method using a graph cut with an estimated trimap. We first estimated the trimap by partitioning the image into three regions: foreground, background and unknown regions. Then the trimap information was incorporated into the graph cut framework by distorting the foreground and background color likelihoods. The proposed algorithm showed good results on the real sequences and also worked at near real-time speed. However, about 60 percent of the processing time of the proposed algorithm was taken from the mean shift segmentation. In future work we are developing an efficient image segmentation method to find proper *automic* regions.

Acknowledgements. This research was supported by the Ministry of Information and Communication, Korea under the Information Technology Research Center support program supervised by the Institute of Information Technology Assessment, IITA-2005-(C1090-0501-0019), and the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei University.

References

1. Y. Boykov, and M. Jolly. Iterative graph cuts for optimal boundary and region segmentation of objects in N-D Images: In: Proc. IEEE Int. Conf. on Computer Vision, (2001) 105-112.
2. Y. Boykov and V. Kolmogorov. An experimental comparision of min-cut/max-flow algorithms for energy minimization in vision: IEEE Trans. on Pattern Anal.and Mach. Intell. 26(9) (2004) 1124-1137.
3. D. Comaniciu, P. Meer: Mean shift: a robust approach toward feature space analysis: IEEE Trans. Pattern Anal. Mach. Intell. 24(5) (2002) 603-619.
4. Y.-Y. Chuang, B. Curless, D. Salesin and R. Szeliski, A bayesian approach to digital matting: In: Proc. Int. Conf. Computer Vison and Pattern Recognition 2 (2001) 264-271.
5. A. Elgmmal, R. Duraiswami, L.S. Davis: Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking: IEEE Trans. Pattern Anal. Mach. Intell. 25(11) (2003) 1499-1504.
6. T. Horprasert, D. Harwood, L.S. Davis: A statistical approach for real-time robust background subtraction and shadow detection. In: Proc. IEEE Frame Rate Workshop (1999) 1-19.
7. M. Harville: A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models. In: Proc. European Conf. on Computer Vision (2002) 543-560.
8. V. Kolmogorov, A. Criminisi, A. Blake, G. Cross and C. Rother: Bi-layer segmentation of binocular stereo video. In: Proc. Int. Conf. on Computer Vision and Pattern Recognition, (2005).
9. Y. Li, J. Sun, C.-K. Tang and H.-Y. Shum: Lazy snapping. ACM Trans. Graphics 23(3) (2004) 303-308.
10. H. Li, M. Greenspan: Multi-scale gesture recognition from time-varying contours. In: Int. Conf. Computer Vision (2005) 236-243.
11. Z. Liu, S. Sarkar: Effect of silhouette quality on hard problems in gait recognition. IEEE Trans. Systems, Man, and Cybernetics-Part B:Cybernetics 35(2) (2005) 170-183.
12. C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts: In: ACM Trans. Graph, 23(3)(2004) 309-314.
13. A. Senior: Tracking people with probabilistic appearance models. In: Proc. IEEE Int. Workshop on PETS, (2002) 48-55.
14. C. Stauffer, W.E.L. Grimson: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 747-757.
15. J. Sun, J. Jia, C.-K. Tang and H.-Y. Shum, Poisson Matting: ACM Transaction on Graphics, 23(3) (2004) 315-321.
16. Y.-L. Tian, M. Lu, A. Hampapur: Robust and efficient foreground analysis for real-time video surveillance. In: Proc. Int. Conf. Computer Vision and Pattern Recognition (2005) 970-975.
17. Z. Tu: An integrated framework for image segmentation and perceptual grouping. In: Int. Conf. Computer Vision (2005) 670-677.
18. L. Wang, T. Tan, H. Ning, W. Hu: Silhouette analysis-based gait recognition for human identification. IEEE Trans. Pattern Anal. Mach. Intell. 25(12) (2003) 1505-1518.

Homography Optimization for Consistent Circular Panorama Generation

Masatoshi Sakamoto[1], Yasuyuki Sugaya[2], and Kenichi Kanatani[1]

[1] Department of Computer Science, Okayama University, Okayama 700-8530, Japan
{sakamoto, kanatani}@suri.it.okayama-u.ac.jp
[2] Department of Information and Computer Sciences,
Toyohashi University of Technology, Toyohashi, Aichi 441-8480 Japan
sugaya@iim.ics.tut.ac.jp

Abstract. A 360° panorama is generated from images freely taken by a hand-held camera: images are pasted together and mapped onto a cylindrical surface. First, the lack of orientation information is overcome by invoking "oriented projective geometry". Then, the inconsistencies arising from a 360° rotation of the viewing direction are resolved by optimizing all the homographies between images simultaneously, using Gauss-Newton iterations based on a Lie algebra representation. The effectiveness of our method is demonstrated using real images.

1 Introduction

A *panorama* is an image with a large angle of view, giving viewers an impression as if they were in front of a real scene. In this paper, we consider a panorama that covers all 360° directions around the viewer. A typical approach for realizing this is to take images with a special optical system such as an fish-eye lens camera, a mirror-based omnidirectional camera, a composite multicamera system, or a camera rotation mechanism [16]. Such optical systems have developed for autonomous robot navigation applications [17].

Another approach is to paste multiple images together, known as *image mosaicing* [13,15]. This technique has been studied in relation to such video image processing as image coding, image compression, background extraction, and moving object detection [2,3,10,12].

However, the aim of this paper is not such industrial or media applications. We consider a situation where travelers take pictures around them using an ordinary digital camera, go home, and create panoramic images for personal entertainment. The purpose of this paper is to present a software system that allows this without using any special device or requiring any knowledge about the camera and the way the pictures were taken.

The principle of image mosaicing is simple. If we find four or more corresponding points between two images, we can compute the *homography* (or *projective transformation*) that maps one image onto the other. Hence, we can warp one image according to the computed homography and past it onto the other image. Continuing this, we can create a panoramic image.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1195–1205, 2006.

However, the images to be pasted distort as we proceed and diverge to infinity when the viewing direction changes by 90°. This can be avoided if we map the images onto a cylindrical surface and unfold it. For this, however, we need to know the orientations of the cameras that took the individual images. That information would be obtained if we used a special device such as an omnidirectional camera or a camera rotation mechanism, but we are assuming that no knowledge is available about camera orientations.

We resolve this difficulty by noting that even though a homography-based panorama cannot be *displayed* on a planar surface beyond a ±90° range, it is nevertheless *mathematically defined* over the entire 360° range. If we invoke the formalism of *oriented projective geometry* [11,14], we can consistently define the pixel values in all directions expressed in homogeneous coordinates, which can then be mapped onto a cylindrical surface.

However, another critical issue arises: if we warp one image and successively paste it onto another, the final image may not agree with the initial image due to accumulated errors. To overcome this, we present a numerical scheme for optimizing all the homographies between images simultaneously subject to the condition that no inconsistency arises. This can be done by using Gauss-Newton iterations based on a Lie algebra representation. We demonstrate the effectiveness of our method using real images.

2 Panorama Generation Using Homographies

2.1 Homographies

As is well known, images taken by a camera rotating around the center of the lens are related to each other by *homographies* (*projective transformations*). This holds for whatever camera motion if the scene is planar or is sufficiently far away. In whichever case, let (x, y) be a point in one image and (x', y') the corresponding point in another. If we represent these by 3-D vectors

$$\boldsymbol{x} = \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{x}' = \begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix}, \tag{1}$$

where f_0 is an arbitrary constant, the homography relationship is written in the form

$$\boldsymbol{x}' = Z[\boldsymbol{H}\boldsymbol{x}]. \tag{2}$$

Here, $Z[\,\cdot\,]$ denotes scale normalization to make the third component 1, and $\boldsymbol{H}$ is a 3×3 nonsingular matrix determined by the relative motion of the camera and its intrinsic parameters. For simplicity, we call the matrix $\boldsymbol{H}$ also a "homography". As Eq. (2) implies, the absolute magnitude and the sign of the matrix $\boldsymbol{H}$ are indeterminate. If we replace the constant f_0 in Eqs. (1) by another value $\tilde{f}_0$, the matrix $\boldsymbol{H}$ in Eq. (2) changes into

$$\tilde{\boldsymbol{H}} = \mathrm{diag}(1, 1, \frac{\tilde{f}_0}{f_0})\boldsymbol{H}\mathrm{diag}(1, 1, \frac{f_0}{\tilde{f}_0}), \tag{3}$$

where diag$(\cdots)$ denotes the diagonal matrix with diagonal elements $\cdots$ in that order).

2.2 Optimal Estimation of Homographies

Equation (2) states that vectors $\boldsymbol{x}'$ and $\boldsymbol{Hx}$ are parallel to each other, so it is equivalently rewritten as

$$\boldsymbol{x}' \times \boldsymbol{Hx} = \boldsymbol{0}. \tag{4}$$

Given N corresponding points (x_α, y_α) and (x'_α, y'_α), $\alpha = 1$, ..., N, between two images, we let $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}'_\alpha$ be their vector representations in the form of Eqs. (1), and $\bar{\boldsymbol{x}}_\alpha$ and $\bar{\boldsymbol{x}}'_\alpha$ their true positions in the absence of noise. If we regard the uncertainty of the x and y coordinates of each point as random Gaussian noise of mean 0 and a constant standard deviation, statistically optimal estimation of the homography $\boldsymbol{H}$ reduces to the minimization of

$$J = \frac{1}{2}\sum_{\alpha=1}^{N} \|\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha\|^2, \tag{5}$$

subject to the constraint

$$\bar{\boldsymbol{x}}'_\alpha \times \boldsymbol{H}\bar{\boldsymbol{x}}_\alpha = \boldsymbol{0}. \tag{6}$$

Eliminating the constraint by introducing Lagrange multipliers and ignoring higher order error terms, we can rewrite Eq. (5) in the following form[1] [6]:

$$J = \frac{1}{2}\sum_{\alpha=1}^{N} (\boldsymbol{x}'_\alpha \times \boldsymbol{Hx}_\alpha, \boldsymbol{W}_\alpha(\boldsymbol{x}'_\alpha \times \boldsymbol{Hx}_\alpha)), \tag{7}$$

$$\boldsymbol{W}_\alpha = \Big(\boldsymbol{x}'_\alpha \times \boldsymbol{HP}_{\mathbf{k}}\boldsymbol{H}^\top \times \boldsymbol{x}'_\alpha + (\boldsymbol{Hx}_\alpha) \times \boldsymbol{P}_{\mathbf{k}} \times (\boldsymbol{Hx}_\alpha)\Big)^-. \tag{8}$$

Here, $(\,\cdot\,)^-$ denotes pseudoinverse, and $\boldsymbol{P}_{\mathbf{k}}$ is the following projection matrix:

$$\boldsymbol{P}_{\mathbf{k}} = \mathrm{diag}(1,1,0). \tag{9}$$

In this paper, we denote by $\boldsymbol{u} \times \boldsymbol{A}$ the matrix whose columns are the vector products of $\boldsymbol{u}$ and the columns of the matrix $\boldsymbol{A}$, and by $\boldsymbol{A} \times \boldsymbol{v}$ the matrix whose rows are the vector products of $\boldsymbol{v}$ and the rows of $\boldsymbol{A}$.

2.3 Panorama Generation

Suppose the user holds a camera roughly horizontally and takes pictures around him roughly at an equal angle. We assume that the scene is sufficiently far away. Since no mechanical device is used, this is merely an approximation. We assume that the focal length is unknown and different from picture to picture.

The user first specifies corresponding points between adjacent images. Various automatic matching techniques have been proposed [8,9,18], but some mismatches are unavoidable using them. In our experiments, we manually selected corresponding points.

[1] The source code of the program that minimizes Eq. (7) by a technique called *renormalization* [7] is available at `http://www.suri.it.okayama-u.ac.jp`

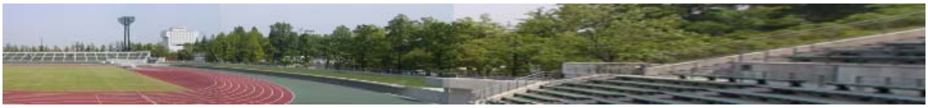


Fig. 1. Successively warping and pasting images using homographies

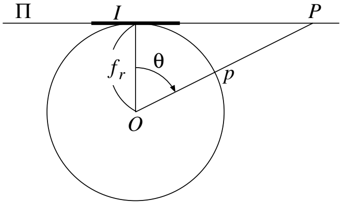


Fig. 2. The rectangular region $\mathcal{I}$ on the tangent plane Π to the cylinder

Then, we compute the homography $\boldsymbol{H}$ between two neighboring images and paste one image onto the other via Eq. (2). Figure 1 is a part of the panoramic image thus created. We can see that images distort more as the viewing direction moves; the distortion diverges to infinity when the viewing direction is orthogonal to its initial orientation.

3 Cylindrical Panorama Generation

3.1 Mapping onto a Cylindrical Surface

The image divergence is avoided if we map the input images onto a cylindrical surface around the viewpoint and unfold it. However, the camera orientation information is missing. This is resolved by invoking *oriented projective geometry* [11,14]. First, we do the following preparation:

- Imagine a hypothetical cylinder of radius f_r around the viewpoint O and define a (θ, h) cylindrical coordinate system (θ around the circumference and h in the axial direction). A point with cylindrical coordinates (θ, h) is unfolded onto a point with Cartesian coordinates $(f_r\theta, h)$.
- Let Π be the plane tangent to the cylinder along the line $\theta = 0$. Define an xy coordinate system on it such that the x-axis coincides with the line $h = 0$.
- Define an xy coordinate system on each input image such that the origin is at the center of the frame with the x-axis extending upward and the y-axis rightward. We identify this xy coordinate system with the xy coordinate system on Π and define on Π a rectangular region $\mathcal{I}$ of the same size as input images centered on $(\theta, h) =$ (0,0) (Fig. 2).
- Number the input images in the order of adjacency, and compute the homography $\boldsymbol{H}_{k(k+1)}$ that maps the kth image onto the $(k+1)$th image from

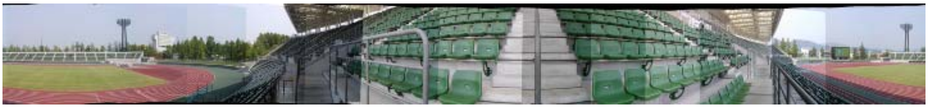


Fig. 3. Circular panorama corresponding to Fig. 1

the specified corresponding points, $k = 1, ..., M$, where we use f_r in the place of f_0 in Eqs. (1) (see Eq. (3)). We choose the sign[2] of $\boldsymbol{H}_{k(k+1)}$ so that $\det \boldsymbol{H}_{k(k+1)} > 0$.

Then, we compute the pixel value of each point p with (discretized) cylindrical coordinates (θ, h) as follows (Fig. 2):

1. Compute the intersection P of the plane $\mathit{\Pi}$ with the line passing through the viewpoint O and the point p on the cylinder, using homogeneous coordinates.
2. If P is inside the region $\mathcal{I}$ and *if the vectors $\vec{Op}$ and $\vec{OP}$ have the same orientation*, copy the pixel value[3] of P in the first image to p.
3. Else, let p' be the point on the cylinder such that[4] $\vec{Op'} \overset{+}{\simeq} \boldsymbol{H}_{12}\vec{Op}$, and compute the intersection P' of $\mathit{\Pi}$ with the line passing through O and p'.
4. If P' is inside the region $\mathcal{I}$ and *if the vectors $\vec{Op'}$ and $\vec{OP'}$ have the same orientation*, copy the pixel value of P in the second image to p.
5. Else, let p'' be the point on the cylinder such that $\vec{Op''} \overset{+}{\simeq} \boldsymbol{H}_{23}\vec{Op'}$, and compute the intersection P'' of $\mathit{\Pi}$ with the line passing through O and p''.
6. If P'' is inside the region $\mathcal{I}$ and *if the vectors $\vec{Op''}$ and $\vec{OP''}$ have the same orientation*, copy the pixel value of P in the third image to p.
7. Repeat the same process over all the images and stop. If no pixel value is obtained, the value of p is undefined.

The radius f_r of the cylinder is arbitrary in principle. However, if we require that the viewing direction should agree with the physical direction, e.g., two viewing directions that make 30° actually make 30° in the scene, the radius f_r should coincide with the focal length f_1 for the first image. This is because the mapping onto the cylinder starts with the first image; the subsequent images are successively warped so as to agree with it. The computation of the focal lengths for all the images will be described in Sec. 4. Figure 3 is a circular panorama thus generated using the images for Fig. 1.

Since the first image is mapped onto the cylinder, next the second image onto the rest of the cylinder, then the third, and so on, images with smaller numbers look "above" images with larger numbers (we can reverse the order, of course).

[2] In practice, it is sufficient if the (33) element is positive. If it is 0, the image origin is mapped to infinity. This does not happen between two overlapping images.

[3] The pixel value of a point with non-integer coordinates is bilinearly interpolated from surrounding pixels.

[4] The relation $\overset{+}{\simeq}$ denotes that one side is a multiple of the other side by a *positive* number.

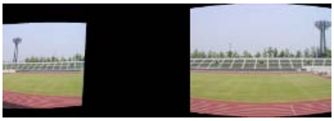


Fig. 4. If neighboring images are successively pasted, the final image does not correctly match the initial image

However, this makes the last image "below" the first image. For consistency, we place the last image above the first image where they overlap (we omit the details).

3.2 Oriented Projective Geometry

In the above procedure, the intersection P of the tangent plane Π is computed in homogeneous coordinates, so no computational failure occurs if P is at infinity.

In the standard projective geometry, a point P on the plane Π is identified with the "line of sight" l passing through P and the viewpoint O (the point P is regarded as located at infinity if the line l is parallel to Π). Since the line of sight l is not oriented, no distinction can be made between "in front of" or "behind" the viewpoint O. It is shown, however, that almost all properties of projective geometry is preserved if the line of sight is oriented [14]. This "oriented projective geometry" was found to be very useful for computer vision applications [11]. In the preceding procedure, we utilized this framework when we signed $\boldsymbol{H}_{k(k+1)}$ so that $\det \boldsymbol{H}_{k(k+1)} > 0$ and compared the "orientations" of the vectors $\vec{Op}$ and $\vec{Op'}$, etc.

4 Simultaneous Homography Optimization

4.1 Discrepancies of the Circular Mapping

If we compute the homographies between two images independently for all pairs, the final image does not necessarily match the initial image correctly, as shown in Fig. 4. This is because due to the accumulation of numerical errors and image distortions, the composite mapping

$$\boldsymbol{H}_{M1}\boldsymbol{H}_{(M-1)M}\boldsymbol{H}_{(M-2)(M-1)}\cdots\boldsymbol{H}_{23}\boldsymbol{H}_{12} \tag{10}$$

does not necessarily define the identity mapping (M is the number of images).

This is not a serious problem if the purpose of the circular panorama is simply for displaying the unfolded image. A more important application is, however, to let the user feel virtual reality by interactively moving the scene as the viewer changes the viewing direction[5]. This can be don by remapping, each the time the

[5] The viewing direction is controlled by a mouse, keyboard, or a joystick. If the user wares a head-mount display (HMD), the head orientation can be measured from the signal it emits.

user specifies the viewing direction, the corresponding part of the cylinder onto its tangent plane[6]. For such applications, no inconsistency is allowed anywhere around the cylinder.

To resolve this, Shum and Szeliski [13] introduced a postprocessing for distributing the inconsistency equally over all image overlaps. Here, we adopt a more consistent approach: we optimize all the homographies simultaneously subject to the constraint that Eq. (10) define the identity map. As a byproduct, the orientations and the focal lengths of the cameras that took the input images are optimally estimated.

4.2 Parameterization of Homographies

While a general homography has 8 degrees of freedom[7], the homography arising from camera rotation and focal length change has only 5 degrees of freedom[8]. If we take the kth image with focal length f, rotate the camera around the lens center by $\boldsymbol{R}$ (rotation matrix), and take the $(k+1)$th image with focal length f', the homography $\boldsymbol{H}_{k(k+1)}$ that maps the kth image onto the $(k+1)$th image has the form

$$\boldsymbol{H}_{k(k+1)} = \mathrm{diag}(1, 1, \frac{f_0}{f_{k+1}})\boldsymbol{R}_{k(k+1)}^{\top}\mathrm{diag}(1, 1, \frac{f_1}{f_0}). \tag{11}$$

Equation (10) defines the identity mapping if and only if

$$\boldsymbol{R}_{12}\boldsymbol{R}_{23}\cdots\boldsymbol{R}_{(M-1)M}\boldsymbol{R}_{M1} = \boldsymbol{I}, \tag{12}$$

where $\boldsymbol{I}$ is the unit matrix. So, we minimize (cf. Eq. (7))

$$J = \frac{1}{2}\sum_{k=1}^{M}\sum_{\alpha=1}^{N_{k(k+1)}}(\boldsymbol{x}_\alpha^{k+1} \times \boldsymbol{H}_{k(k+1)}\boldsymbol{x}_\alpha^k, \boldsymbol{W}_\alpha^{k(k+1)}(\boldsymbol{x}_\alpha^{k+1} \times \boldsymbol{H}_{12}\boldsymbol{x}_\alpha^k)), \tag{13}$$

subject to the constraint that all $\boldsymbol{H}_{k(k+1)}$ have the form of Eq. (11) in such a way that Eq. (12) is satisfied. In Eq. (13), we assume that $N_{k(k+1)}$ points $\boldsymbol{x}_\alpha^k$ in the kth image correspond to the $N_{k(k+1)}$ points $\boldsymbol{x}_\alpha^{k+1}$ in the $(k+1)$th image, and the subscript k is computed modulo M. The matrix $\boldsymbol{W}_\alpha^{k(k+1)}$ is the value of $\boldsymbol{W}$ in Eq. (8) for the kth and the $(k+1)$th images.

Through Eq. (11), the function J in Eq. (13) is regarded as a function of the focal lengths f_1, f_2, ..., f_M and the rotations $\boldsymbol{R}_{12}$, $\boldsymbol{R}_{23}$, ..., $\boldsymbol{R}_{M1}$; we minimize J with respect to them subject to Eq. (12).

[6] A well known such system is QuickTime VR [1], for which input images are taken using a special camera rotation mechanism.

[7] The 3 × 3 matrix $\boldsymbol{H}$ has 9 elements, but there is an overall scale indeterminacy.

[8] The camera rotation matrix $\boldsymbol{R}$ has 3 degrees of freedom, to which are added the focal lengths f and f' before and after the camera rotation.

4.3 Lie Algebra Approach

Each rotation $\boldsymbol{R}_{k(k+1)}$ is specified by three parameters, but using a specific parameterization such as the Euler angles θ, ϕ, and ψ complicates the equation. So, we adopt the well known method of Lie algebra [4,5]. Namely, instead of directly parameterizing $\boldsymbol{R}_{k(k+1)}$, we specify its "increment[9]" in each step. To be specific, we exploit the fact that a small change of rotation $\boldsymbol{R}_{k(k+1)}$ has the following form [4,5]:

$$\boldsymbol{R}_{k(k+1)} + \boldsymbol{\omega}_{k(k+1)} \times \boldsymbol{R}_{k(k+1)} + \cdots. \tag{14}$$

Here, $\cdots$ denotes terms of order 2 or higher in $\boldsymbol{\omega}_{k(k+1)}$. If we replace $\boldsymbol{R}_{k(k+1)}$ in $\boldsymbol{H}_{k(k+1)}$ by Eq. (14), Eq. (13) can be regarded as a function of f_1, ..., f_M and $\boldsymbol{\omega}_{12}$, ..., $\boldsymbol{\omega}_{M1}$. After minimizing it with respect to them, the rotations $\boldsymbol{R}_{k(k+1)}$ are updated by

$$\boldsymbol{R}_{k(k+1)} \leftarrow \mathcal{R}(\boldsymbol{\omega}_{k(k+1)})\boldsymbol{R}_{k(k+1)}, \tag{15}$$

where $\mathcal{R}(\boldsymbol{\omega})$ denotes the rotation around axis $\boldsymbol{\omega}$ by angle $\|\boldsymbol{\omega}\|$.

The derivatives of J with respect to f_1, ..., f_M and $\boldsymbol{\omega}_{12}$, ..., $\boldsymbol{\omega}_{M1}$ can be analytically calculated (we omit the details). If we introduce the Gauss-Newton approximation, we can also calculate the second derivatives in simple analytic forms (we omit the details). It seems, therefore, that we can minimize J by Gauss-Newton iterations. However, there is a serious difficulty in doing this.

4.4 Alternating Optimization Approach

We need to enforce the constraint of Eq. (12). To a first approximation, Eq. (12) is expressed in the following form (we omit the details):

$$\boldsymbol{\omega}_{12} + \boldsymbol{R}_{12}\boldsymbol{\omega}_{23} + \boldsymbol{R}_{12}\boldsymbol{R}_{23}\boldsymbol{\omega}_{34} + \cdots + \boldsymbol{R}_{12}\boldsymbol{R}_{23}\cdots\boldsymbol{R}_{(M-1)M}\boldsymbol{\omega}_{M1} = \boldsymbol{0}. \tag{16}$$

A well known strategy for constrained optimization is the *method of projection*: the parameters are incremented without considering the constraint and then projected onto the constraint surface in the parameter space. However, if we try to minimize Eq. (13) without considering Eq. (16), *the solution is indeterminate* (the Hessian has determinant 0).

We resolve this difficulty by adopting the *alternate optimization*. Namely, Eq. (13) is first minimized with respect to f_1, ..., f_M with $\boldsymbol{R}_{12}$, ..., $\boldsymbol{R}_{M1}$ fixed. Then, the result is minimized with respect to $\boldsymbol{\omega}_{12}$, ..., $\boldsymbol{\omega}_{M1}$ with f_1, ..., f_M fixed. This time, the solution is unique because the Hessian of J with respect to $\boldsymbol{\omega}_{12}$, ..., $\boldsymbol{\omega}_{M1}$ alone is nonsingular.

Next, Eq. (16) is imposed by projection in the form of

$$\hat{\boldsymbol{\omega}}_{k(k+1)} = \boldsymbol{\omega}_{k(k+1)} - \Delta\boldsymbol{\omega}_{k(k+1)}. \tag{17}$$

[9] The linear space defined by such (mathematically infinitesimal) increments is called the *Lie algebra so*(3) of the group of rotations $SO(3)$ [4].

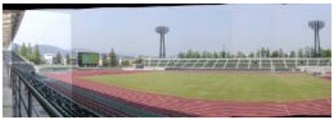


Fig. 5. After the simultaneous optimization of homographies, the discrepancy in Fig. 4(b) disappears

The correction $\Delta\boldsymbol{\omega}_{k(k+1)}$ is determined as follows. The condition for Eq. (17) to satisfy Eq. (16) is written as

$$\boldsymbol{S}\Delta\tilde{\boldsymbol{\omega}} = \boldsymbol{e}, \tag{18}$$

where we define

$$\begin{aligned} \boldsymbol{S} &= \left(\boldsymbol{I}\ \boldsymbol{R}_{12}\ \boldsymbol{R}_{12}\boldsymbol{R}_{23} \cdots \boldsymbol{R}_{12}\boldsymbol{R}_{23} \cdots \boldsymbol{R}_{(M-1)M} \right), \\ \Delta\tilde{\boldsymbol{\omega}} &= \left(\Delta\boldsymbol{\omega}_{12}^\top\ \Delta\boldsymbol{\omega}_{23}^\top \cdots \Delta\boldsymbol{\omega}_{M1}^\top \right)^\top, \\ \boldsymbol{e} &= \boldsymbol{\omega}_{12} + \boldsymbol{R}_{12}\boldsymbol{\omega}_{23} + \boldsymbol{R}_{12}\boldsymbol{R}_{23}\boldsymbol{\omega}_{34} + \cdots + \boldsymbol{R}_{12}\boldsymbol{R}_{23} \cdots \boldsymbol{R}_{(M-1)M}\boldsymbol{\omega}_{M1}. \end{aligned} \tag{19}$$

Introducing Lagrange multipliers, we can obtain the solution $\Delta\tilde{\boldsymbol{\omega}}$ that minimizes $\|\Delta\tilde{\boldsymbol{\omega}}\|$ subject to Eq. (18) in the form

$$\Delta\tilde{\boldsymbol{\omega}} = \boldsymbol{S}^\top(\boldsymbol{S}\boldsymbol{S}^\top)^{-1}\boldsymbol{e}. \tag{20}$$

From this, the correction formula of Eq. (17) reduces to the following form (we omit the details):

$$\hat{\boldsymbol{\omega}}_{12} = \boldsymbol{\omega}_{12} - \frac{1}{M}\boldsymbol{e},\ \ \hat{\boldsymbol{\omega}}_{23} = \boldsymbol{\omega}_{23} - \frac{1}{M}\boldsymbol{R}_{12}^\top\boldsymbol{e},\ \ \hat{\boldsymbol{\omega}}_{34} = \boldsymbol{\omega}_{34} - \frac{1}{M}\boldsymbol{R}_{12}^\top\boldsymbol{R}_{23}^\top\boldsymbol{e},$$
$$\ldots,\quad \hat{\boldsymbol{\omega}}_{M1} = \boldsymbol{\omega}_{M1} - \frac{1}{M}\boldsymbol{R}_{12}^\top\boldsymbol{R}_{23}^\top \cdots \boldsymbol{R}_{(M-1)M}^\top\boldsymbol{e}. \tag{21}$$

The rotations $\boldsymbol{R}_{12}$, ..., $\boldsymbol{R}_{M1}$ are updated by Eq. (15). With these fixed, Eq. (13) is again minimized with respect to f_1, ..., f_M, and the same procedure is iterated.

Since Eq. (16) is a first approximation in $\boldsymbol{\omega}_{k(k+1)}$, the rotations $\boldsymbol{R}_{k(k+1)}$ updated by Eq. (15) may not strictly satisfy Eq. (12). However, the discrepancy is very small, so we randomly choose one rotation matrix and replace it by the value that strictly satisfies Eq. (15), i.e., replace it by the inverse of the product of the remaining rotation matrices.

Figure 5 is the result corresponding to Fig. 4. No discrepancy occurs this time. This can be confirmed by many examples, which are not shown here due to page limitation, though.

5 Concluding Remarks

We have shown a consistent technique for generating a 360° circular panorama from images taken by freely moving a hand-held camera.

The first issue is the lack of camera orientation information, which was resolved by computing the homographies between neighboring images and invoking the framework of "oriented projective geometry". The second issue is the discrepancies between the final image and the initial image due to accumulated errors. We resolved this by simultaneously optimizing all the homographies. To this end, we introduced the Gauss-Newton iterations using the Lie algebra representation and the alternating optimization scheme.

In practice, we need to add further corrections and refinements for eliminating intensity discontinuities and geometric discrepancies at individual image boundaries. This can be done easily using existing techniques (we omit the details).

References

1. S.E. Chen, QuickTime VR—An image-based approach to virtual environment navigation, *Proc. SIGGRAPH'95*, August 1995, Los Angeles, U.S.A., pp. 29–38.
2. M. Irani, S. Hsu, and P. Anandan, Video compression using mosaicing representations, *Signal Process: Image Comm.*, **7**-4/5/6 (1995-11), 529–552.
3. M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, Efficient representations of video sequences and their applications, *Signal Process: Image Comm.*, **8**-4/5/6 (1996-11), 327–351.
4. K. Kanatani, *Group-Theoretical Methods in Image Understanding*, Springer, Berlin, Germany, 1990.
5. K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, U.K., 1993.
6. K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam, the Netherlands, 1996; reprinted, Dover, New York, NY, U.S.A., 2005.
7. K. Kanatani, N. Ohta and Y. Kanazawa, Optimal homography computation with a reliability measure, *IEICE Trans. Inf. & Syst.*, **E83-D**-7 (2000-7), 1369–1374.
8. Y. Kanazawa and K. Kanatani, Robust image matching preserving global consistency, *Proc. 6th Asian Conf. Comput. Vision*, January 2004, Jeju, Korea, Vol. 2, pp.1128–1133.
9. Y. Kanazawa and K. Kanatani, Image mosaicing by stratified matching, *Image Vision Computing*, **22**-2 (2004-2), 93–103.
10. M.-C. Lee, C.-l. B. Lin, C. Gu, T. Markoc, S I. Zabinski, and R. Szeliski, A layered video object coding system using sprite and affine motion model, *IEEE Trans. Circuits Video Tech.*, **7**-1 (1997-2), 130–145.
11. S. Leveau and O. Faugeras, Oriented projective geometry for computer vision, *Proc. 4th Euro. Conf. Comput. Vis.*, Vol. 1, April 1996, Cambridge, U.K., pp. 147–156.
12. H.S. Sawhney and S. Ayer, Compact representations of videos through dominant and multiple motion estimation, *IEEE Trans. Patt. Anal. Machine Intell.*, **18**-8 (1996-4), 814–830.
13. H.-Y. Shum and R. Szeliski, Construction of panoramic image mosaics with global and local alignment, *Int. J. Comput. Vis.*, **36**-2 (2000-2), 101–130.
14. J. Stolfi, *Oriented Projective Geometry: A Framework for Geometric Computation*, Academic Press, San Diego, CA, U.S.A., 1991.
15. R. Szeliski and H.-U. Shum, Creating full view panoramic image mosaics and environment maps, *Proc. SIGGRAPH'97*, August 1997, Los Angeles, CA, U.S.A., pp.251–258.

16. Y. Yagi, Omnidirectional sensing and its applications, *IEICE Trans. Inf. & Syst.*, **E82-D**-3 (1999-3), 568–579.
17. Y. Yagi, K. Imai, K. Tsuji and M. Yachida, Iconic memory-based omnidirectional route panorama navigation, *IEEE Trans. Patt. Anal. Mach. Intell.*, **27**-1 (2005-1), 78–87.
18. Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, *Artif. Intell.*, **78**-1/2 (1995-10), 87–119.

Graph-Based Global Optimization for the Registration of a Set of Images

Hui Zhou

Epson Edge, Epson Canada Ltd.,
3771 Victoria Park Ave., Toronto, Ontario, Canada M1W 3Z5
Hui_Zhou@ea.epson.com

Abstract. In this paper, we introduce a fast and efficient global registration technique in photo-stitching, in which, the layout of the set of images has multiple rows and columns. The proposed algorithm uses a graph based model to deal with the problem of global registration. By using alternative path from the graph containing the image layout, it is possible to align the images against the reference image when pair-wise registration fails.

Keywords: Global registration; panoramic imaging; image mosaics.

1 Introduction

With the popularity of digital cameras, it has been an active field for researchers and commercial practitioners to create mosaics and panoramic images by stitching multiple photos or video frames. Numbers of approaches [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] have been proposed. There are usually two ways to generate the image mosaic: *frame-to-mosaic* and *frame-to-frame*. Frame-to-mosaic is to align each new frame to the mosaic being built, thus the registration happens between each frame and the mosaic being formed; in another way, frame-to-frame means to acquire the transform between frames through registration and then map all of them to the mosaic. For the first approach, it is obvious that the registration is working between original frame and transformed frame (the mosaic). Hence, it works fine only when there is sufficient overlap between frames. The second approach can work on the large view angle case. However, since the registration is done in a pairwise way locally, small registration error between frames can be accumulated, and amplified when all the frames are aligned together as one mosaic.

One solution to overcome the problem in frame-to-frame case is to apply a global optimization step such that all the frame-to-mosaic mappings are maximally consistent with all the local pairwise registration. The standard optimization technique is bundle adjustment, derived from photogrammetry [11]. Researchers have proposed global optimization techniques based on this technique. For example, Szeliski and Shum [5] proposed a global optimization based on bundle adjustment, in which the minimization is formulated as a constrained least-squares problem with the 3D rotational and focal length model. In [4], Sawhney et al proposed a framework, in which topology and registration are combined. The global alignment is done through

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1206 – 1214, 2006.

a way similar as bundle adjustment approach. Xiong et al [6] used simulated annealing technique for global optimization.

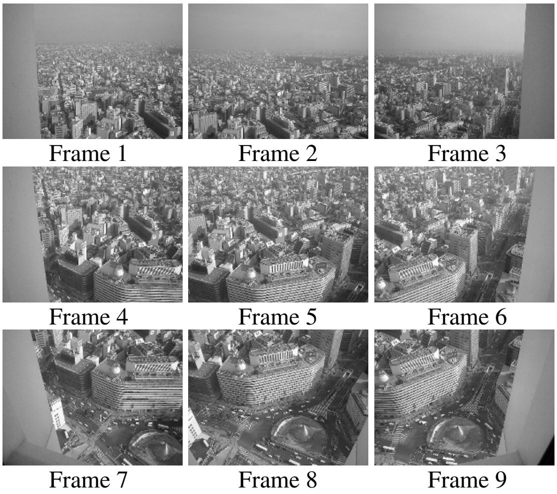
This paper proposes a novel algorithm used for the optimization of registration for a set of images with multiply rows and multiple columns. The algorithm aims to estimate the optimal transform parameters for each frame against with the reference frame such that the stitched output result contains the reference frame on which no transform is needed to apply. The *registration graph* is introduced and built through registration steps. By using registration graph, the proposed approach tries to optimize the registration result -- to replace failed registered pairs by registering other pairs; and reduce the registration error globally.

In the paper, we assume that the layout of the image set is defined based on 4-neighbor connectivity, i.e., a frame could have overlapped area with its up, down, left and right neighbors. Note that it is not necessary for any image to have all 4 overlapped neighboring images. However, any image should have at least one connected neighbors, i.e., no isolated images.

Pairwise registration is the base for the proposed optimization algorithm. It has been well explored by researchers [12]. In this paper, feature-based registration technique [13] is used for the registration of a pair of frames. The result of each pairwise registration is then the matching points list for the two images.

2 Proposed Algorithm

We start the registration from the reference image and its four neighbors. The reference image can be specified as any one of the images. As the example shown in Figure 1, we can suppose the reference image is frame 5.



Frame 1 Frame 2 Frame 3

Frame 4 Frame 5 Frame 6

Frame 7 Frame 8 Frame 9

Fig. 1. An example of 3x3 image sequence. The resolution has been reduced for illustration purpose (original resolution 1360x1024 for each frame).

2.1 Registration Cycle

The first cycle of registration is between frame 5 and its four neighbors, as illustrated in Figure 2.

In the following of the paper, we use number with prefix of 'F' to note the frame. From Figure 2A, it is obvious that four times pairwise registration is needed: F2 against F5; F4 against F5; F6 against F5; and F8 against F5. After the first cycle of registration, we will continue the registration for those frames further to the reference frame. To avoid the misalignment error between unregistered images pairs, we align one unregistered frame with its two successfully registered neighboring frames simultaneously, as illustrated in Figure 2B. Next section will elaborate as how to align one frame against more than one frame.

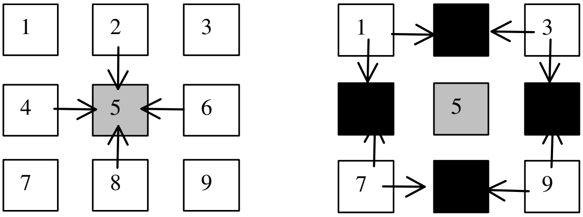


Fig. 2. Left (A): The 4 registrations between the reference (F5) and its four neighbors. Gray block indicates the reference frame, while white as unregistered frames. Right (B): The 2nd registration circle in which the frames will be aligned against the successfully registered frames (black blocks) in the previous cycle.

The registration cycles go on in a way that frames closer to the reference frame have higher priority than those further to the reference frame.

2.2 Registering One Against More Than One Image

The motivation to register one frame against its two successfully registered neighbors, rather than one of them, is to obtain better estimation by taking advantage of the fact that the frame shares overlapping region both with its register-able neighbors. As the example shown in Figure 2B, if we only align frame 1 against frame 2 in the registration, there could be a big misalignment between frame 1 and frame 4 when we finally put all the frames to align with the reference frame.

In this paper, the implementation of registering one frame against more than one frame is done by solving augmented over-constrained problems. Solving the over-constrained linear equations is the same as the solution described in reference [1]. For example, suppose we have frame F1, we need to align it with other two frames, F2 and F4. We can do the following steps to align F1 with both F2 and F4:

1) Pairwise registration for F1 and F2. If succeeds, obtain matching points list between F1 and F2 and set it to Q1; otherwise quit.

2) Pairwise registration for F1 and F4. If succeeds, obtain matching points list between F1 and F4 and set it to Q2; otherwise quit.

3) Combine Q1 and Q2 to Q. Set up a set of over-constrained linear equations based on Q, and solve it.

2.3 Registration Graph

As the registration continues cycle by cycle, we construct the registration graph to record the registration information between frames. A *registration graph* is a directed graph representation, in which each vertex indicates a frame, and the edge between vertices indicates the alignment. The registration graphs shown in Figure 3 are the graphs we constructed in the different status for the set of sample images shown in Figure 1. Pairwise registration fails on F4 against F5 and F6 against F5, but succeeds on F2 against F5 and F8 against F5, as shown in Status 1. This registration graph is updated after every registration cycle finished. It should be mentioned that in the registration cycles, the frame will be registered only against its previously registerable neighbors. For example, in the 2nd registration cycle, for frame 1, we only register it against F2, rather than against F2 and F4, this is because F4 failed in the previous registration cycle. The same situation happens for F3 and F7. Figure 3 Status 2 shows the registration graph after all the registration cycles finishes. Apparently, not all images are successfully registered at this time.

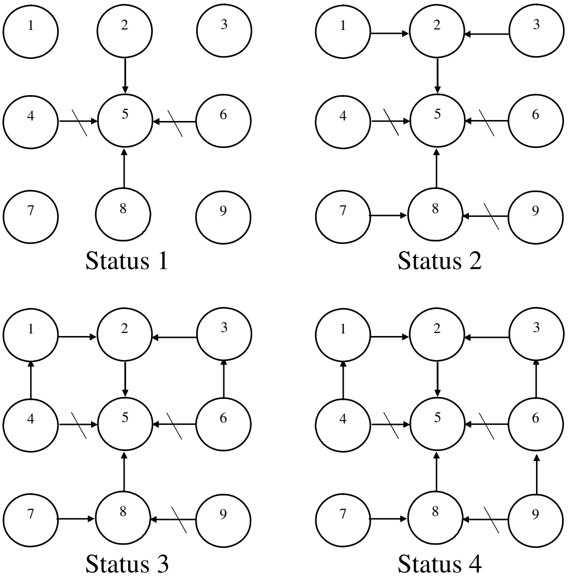


Fig. 3. Illustration of the construction of the registration graph. An arrow shows successful pairwise registration, and an arrow with a backslash on means pairwise registration fails on the pair.

2.4 Re-registration

For the frames failed to be registered after all registration cycles end, we will do registration again to align it against one of its neighbors from which can form an alternative path leading to the reference frame. For example, for the graph shown in Figure 3 Status 2, F4 failed to be aligned to reference frame (F5), we should, however, be able to align it with the reference if we can register it against Frame 1, which has been successfully registered in the 2nd registration cycle.

But how to find the proper neighbor to form the path from the current frame to the reference frame? In the example shown in Figure 3 Status 2, we could potentially choose F4 -> F1 -> F2 -> F5, or F4 -> F7 -> F8 ->F5. In this paper, a shortest-path algorithm is applied to find the proper neighbors. The reason for finding the shortest path is from the observation that the shorter the path is the smaller the accumulated registration error along the path is.

Floyd's shortest-path finding algorithm is used to find the shortest path in the registration graph. For the example shown in Figure 3 Status 2, path F4 -> F1 -> F2 -> F5 and F4 -> F7 -> F8 -> F5 are both the shortest ones. We only need to pick one of them, say, F4 -> F1 -> F2 -> F5. We also can obtain another shortest-path when trying to align frame 6 to the reference: F6 -> F3 -> F2 -> F5. After we find the shortest path and determine the proper neighbor of the unregistered frame, we will register the frame with its neighbor and modify the registration graph if registration succeeds. The example shown in Figure 3 Status 2 becomes the one shown in Figure 3 Status 3 after we successfully register F4 against F1 and F6 against F3.

It can be noticed that F9 is still not successfully aligned to the reference in Figure 3 Status 3. We can get it by repeat the process described above: find the path F9 -> F6 -> F3 -> F2 -> F5, then register F9 -> F6, and modify the registration graph as shown in Figure 3 Status 4.

We repeatedly do this process: shortest-path finding, determining the proper neighbor for registration and modifying the registration graph until every frame has been successfully registered or there is no way to align unregistered frames.

2.5 Aligning to the Reference with Less Error

By multiplying the transform matrices along the shortest path in the registration graph from any frame to the reference frame, we could obtain the direct transform matrix between each frame and the reference frame. For example, The transform matrix beween Frame 1 against Frame 5 $M_{[1][5]}$ should be obtained by $M_{[1][5]} = M_{[1][2]} \bullet M_{[2][5]}$, where $M_{[1][2]}$ and $M_{[2][5]}$ are the estimated transform matrices between F1 and F2 and F2 and F5, respectively.

The transform matrices obtained by multiplying the matrices along the path to the reference may not be exactly accurate. Suppose we have a very small error in the transform matrix between each pair:

$M_{[1][5]} = \hat{M}_{[1][5]} + M_{\delta 15}$; $M_{[1][2]} = \hat{M}_{[1][2]} + M_{\delta 12}$ and

$M_{[2][5]} = \hat{M}_{[2][5]} + M_{\delta 25}$; where, $\hat{M}_{[1][2]}$, $\hat{M}_{[2][5]}$ and $\hat{M}_{[1][5]}$ are the ideal, correct transform matrices and $M_{[1][2]}$, $M_{[2][5]}$ and $M_{[1][5]}$ are the estimated transform matrices between Frame 1 and 2, Frame 2 and 5, and Frame 1 and 5, respectively. $M_{\delta 15}$, $M_{\delta 12}$ and $M_{\delta 25}$ are the corresponding error matrices between correct matrices and estimated ones. By Multiplying them, we get

$\hat{M}_{[1][5]} + M_{\delta 15} = (\hat{M}_{[1][2]} + M_{\delta 12})(\hat{M}_{[2][5]} + M_{\delta 25})$ while $\hat{M}_{[1][5]} = \hat{M}_{[1][2]}\hat{M}_{[2][5]}$, then

$M_{\delta 15} = M_{[1][2]}\hat{M}_{\delta 25} + M_{\delta 12}\hat{M}_{[2][5]} + M_{\delta 12}M_{\delta 25}$

Therefore, through matrix multiplication, the error might be accumulated and even amplified. This accumulated error becomes even larger when the multiplication sequence is longer (which is the case that the registration path is longer). Even we have used the shortest path to reduce this accumulated error; the error can be relatively big.

We therefore propose an algorithm to reduce the effect of the cumulative error. During the transformation of a registered image, the matching point list along the registration path from the registered image to the reference image is remapped. In the example illustrated in Figure 5, the frame 1 is registered relative to previously-registered frame 2, which in turn, is registered to the reference frame 5. A point P in frame 1 corresponds to a point Q in frame 2, and point R in frame 2 corresponds to a point S in reference frame 5.

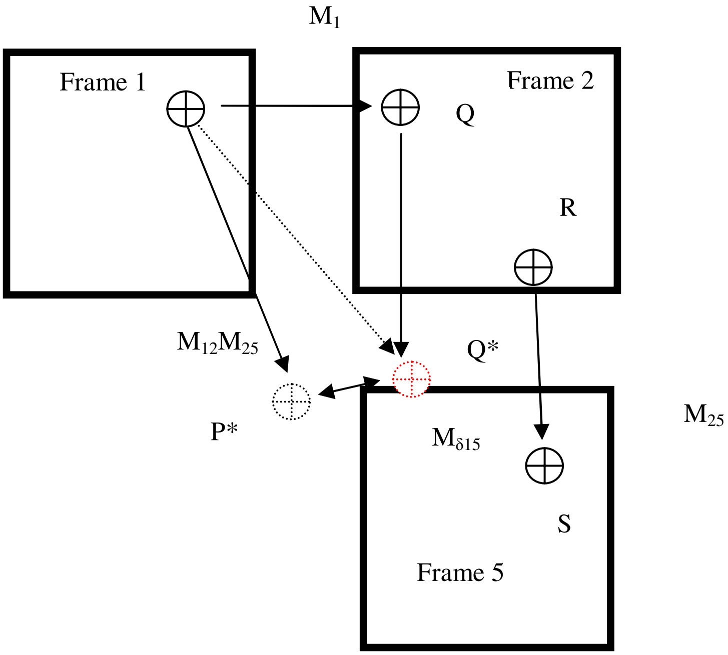


Fig. 4. Correction of the accumulated error resulting from multiplication of transform matrices

The transform matrices $M_{[1][2]}$ between F1 and F2, and $M_{[2][5]}$ between images F2 and F5 are estimated by solving the corresponding matching point lists. A point Q^* corresponding to the point Q after having been transformed to the reference image using $M_{[2][5]}$ can be calculated. A point P^* corresponding to the point P after having been transformed to the reference image using $M_{[2][5]}\bullet M_{[1][2]}$ can also be calculated. It should be noted that points P^* and Q^* can be transformed to locations inside or outside of the reference image I_5.

Ideally, P^* should be located at the same point as Q^*. This is not, however, typically the case. P^* can differ from Q^* as Q^* is calculated from $M_{[2][5]}$, whereas P^*

is calculated using $M_{[2][5]} \bullet M_{[1][2]}$. As noted above, a cumulative error can result from one or more matrix multiplications. As a result, it is obvious that Q^* should be more accurate than P^*. The transform matrix $M_{[1][5]}$ can then be corrected to $\hat{M}_{[1][5]}$ by determining the transformation between the feature points in F1 and the corresponding transformed (i.e., transformed to F5) feature points from F2, thereby cancelling the additional error present in $M_{[1][5]}$ determined using the multiplied individual transformations. This correction is repeated for all registration paths containing three or more images.

For illustration purposes, assume that, for a path from image I_1 to image I_2 to … image I_N, image I_1 is to be aligned to image I_N. P_i and P_j are coordinates of matched points in images I_i and I_j, respectively. The transform matrix $M_{[1][N]}$ can be adjusted to alleviate the effect of the cumulative error in an iterative way by using the following approach:

```
1) $i \leftarrow N-2,\ j \leftarrow i+1$
2) $P_j' \leftarrow M_{[j][N]} P_j$
3) Solve $M_{[i][N]} P_i = P_j'$ to get $M_{[i][N]}$
4) $i \leftarrow i-1,\ j \leftarrow i+1$
5) If $i \geq 1$, go to step 2; otherwise end of the algorithm.
```

After the alignment to reference frame and the correction of accumulated error, all successfully registered frames in previous registratioin cycles and re-registration steps have been aligned to the reference frame. Each frame now has the estimated transform matrix, with which can be used to transfrom the frame onto the reference plane.

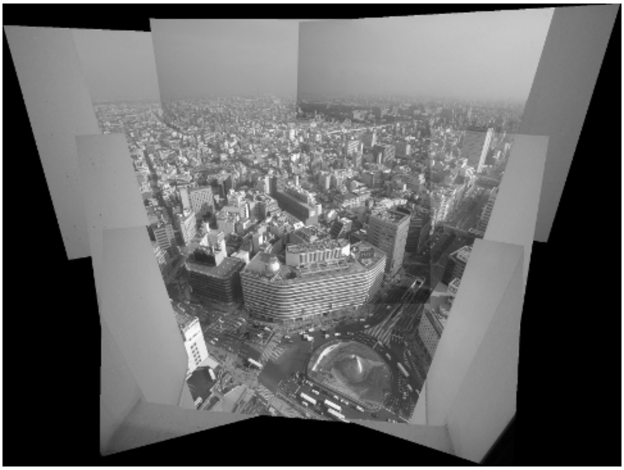


Fig. 5. Composed output result by transforming all frames against the reference frame

3 Experiment Result

To show the registration result, we transformed each frame in Figure 1 using the corresponding estimated transform matrices. When generating the output composition result, as shown in Figure 5, one frame is simply put on the top of another. No blending in the overlapping region is applied. The output images were generated from original sample images, but the picture shown in Figure 5 has been resized for the illustration purpose. On PC with 1.7GHz CPU and 512MB memory running Microsoft Windows 2000, the total registration time for this 3x3 example is 2.7 seconds.

4 Concluding Remarks

This paper proposed a graph based optimization technique for image sequence registration. By introducing registration graph and finding the shortest path, the algorithm is able to provide more accurate and robust registration for image sequence. Accumulated registration error along the path to reference can be further reduced from the correction step.

References

1. Szeliski, R. and Shum, H. Creating Full View Panoramic Mosaics and Environment Maps. SIGGRAPH, (1997) 251-258
2. Peleg. S. and Herman, J. Panoramic Mosaics by Manifold Projection. CVPR, (1997) 338-343
3. Irani, M., Anandan, P. and Hsu, S. Mosaic based representations of video sequences and their applications. ICCV, (1995) 605-611
4. Sawhney, H. S., Hsu, S., and Kumar, R. Robust video mosaicing through topology inference and local to global alignment. ECCV, (1998) 103- 119
5. Shum. H. and Szeliski, R. Construction and refinement of panoramic mosaics with global and local alignment. ICCV, (1998) 953-958
6. Xiong, Y. and Turkowski, K. Registration, calibration and blending in creating high quality panoramas. Proceedings of Applications of Computer Vision, (1998) 69-74
7. Krishnan, A. and Ahuja, N., Panoramic Image Acquisition, Proceedings IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, June 18-20, (1996) 379-384
8. R. Benosman, S. B. Kang, and O. Faugeras, Eds., Panoramic Vision, Springer-Verlag, (2001)
9. Chu-Song Chen, Yu-Ting Chen, Fay Huang: Stitching and Reconstruction of Linear-Pushbroom Panoramic Images for Planar Scenes. ECCV (2) (2004) 190-201
10. Tomio Echigo, Richard J. Radke, Peter J. Ramadge, Hisashi Miyamori, Shun-ichi Iisaku: Ghost Error Elimination and Superimposition of Moving Objects in Video Mosaicing. ICIP (4) (1999) 128-132

11. Slama C. C. (Ed) Manual of Photogrammetry. American Society of Photogrammetry and Remote Sensing, Virginia, USA, (1980)
12. Brown, L.G. A survey of image registration techniques. ACM Computing Surveys, 24 (1992) 326–376
13. Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.-T. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. Artificial Intelligence Journal, 78, (1995) 87-119

One-Dimensional Search for Reliable Epipole Estimation

Tsuyoshi Migita and Takeshi Shakunaga

Okayama University, 3-1-1, Tsushima-naka, Okayama, Japan
{migita, shaku}@chino.it.okayama-u.ac.jp

Abstract. Given a set of point correspondences in an uncalibrated image pair, we can estimate the fundamental matrix, which can be used in calculating several geometric properties of the images. Among the several existing estimation methods, nonlinear methods can yield accurate results if an approximation to the true solution is given, whereas linear methods are inaccurate but no prior knowledge about the solution is required. Usually a linear method is employed to initialize a nonlinear method, but this sometimes results in failure when the linear approximation is far from the true solution. We herein describe an alternative, or complementary, method for the initialization. The proposed method minimizes the algebraic error, making sure that the results have the rank-2 property, which is neglected in the conventional linear method. Although an approximation is still required in order to obtain a feasible algorithm, the method still outperforms the conventional linear 8-point method, and is even comparable to Sampson error minimization.

Keywords: fundamental matrix, epipole estimation, polynomial equation, resultant.

1 Introduction

Given a pair of images, and a set of correspondences of the feature points in the images, we can estimate the epipolar geometry, or the fundamental matrix, which can be used in reconstruction of the 3D geometry of the cameras and that of the scene [1, 2]. Epipolar geometry estimation is one of the most important issues in computer vision, and a great number of studies have investigated solutions to epipolar geometry estimation. Although a certain class of the problem could be solved easily by even a naive method, another class was found to be very sensitive to the method used [1, 3]. Therefore, a sophisticated method that has a wider range of application is required. The present paper describes a new approach that reduces the estimation to a high order polynomial equation in a single variable, which enables a reliable initial estimation for the best estimation method in the literature.

The most accurate results are obtained by *maximum likelihood* (ML) estimation [4], which minimizes the squared sum of reprojection errors by estimating the position, the orientation and the intrinsic parameters for each camera, as well as the 3D position for each feature point. In addition, this method can easily

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1215–1224, 2006.

be extended to deal with an arbitrary camera model (e.g. with lens distortion), as well as an arbitrary noise model (e.g. removing outliers). A more convenient method estimates a fundamental matrix that minimizes the *Sampson error* [1,3], and it is known that the results are sufficiently accurate for most cases. However, both methods require a reliable initial estimation for an iterative process of nonlinear optimization to be successful.

The well-known *linear eight-point algorithm* [5] (hereinafter L8P) could be regarded as a further approximation to the Sampson error minimization. The L8P algorithm requires two approximations to be performed only with linear algebra. First, the cost function should be approximated to be a bilinear form of the nine elements of the fundamental matrix. Second, in the main calculation, it cannot constrain the resulting matrix to be of rank 2, and the constraint is enforced afterward. Since this method can generate a result without any prior knowledge, it is usually used as an initial estimation for one of the more accurate nonlinear methods, such as those described above. However, this may fail because the L8P can only produce a *sub-optimal* result, due to the approximations described above.

Therefore, it would be beneficial to construct another method that could complement L8P. For this purpose, in the present paper, we reduce the estimation to a high order polynomial equation in a single variable using the resultant, which is a mathematical tool for eliminating variables from a nonlinear equation system. For a single-variable polynomial equation, we can enumerate all of the possible solutions without any initial estimation, whereas L8P can obtain only one candidate. We herein derive two methods, an exact method and its approximation. The exact method minimizes the same cost function as that of the L8P, but the rank-2 constraint is fully taken into account. Therefore, in principle, the results are more accurate than L8P. Since the exact method requires an extreme amount of computation, we need an approximation. However, the approximation method still outperforms L8P, and the results are comparable to those of Sampson error minimization.

2 Mathematical Background

In this section, we briefly introduce several mathematical concepts that are used extensively throughout the present paper.

Consider an $n \times n$ matrix A and an n-vector $\boldsymbol{x}$ satisfying $A\boldsymbol{x} = 0$. The determinant of A is 0, and $\boldsymbol{x}$ is the right null vector of A. The elements of matrix A can be polynomials, and then the determinant is also a polynomial.

Consider two polynomials $P = \sum_{i=0}^{l} p_i x^i$ and $Q = \sum_{i=0}^{m} q_i x^i$ and observe that the following equation holds when $P = Q = 0$,

$$\begin{bmatrix} p_0 & p_1 & \cdots & p_l & & \\ & p_0 & p_1 & \cdots & p_l & \\ & & \ddots & & & \ddots \\ q_0 & q_1 & \cdots & q_m & & \\ & q_0 & q_1 & \cdots & q_m & \\ & & \ddots & & & \ddots \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{l+m} \end{bmatrix} = 0 \tag{1}$$

where, in the matrix, the upper m rows contain p_i's, and the lower l rows contain q_i's. The matrix is of size $(l+m) \times (l+m)$, and its determinant must be 0 if and only if $P = Q = 0$. In addition, the right null space of the matrix contains a geometric series of x. Such a matrix is called the Sylvester matrix, and its determinant is the resultant of the polynomials P and Q, which we can be written as $\mathrm{Res}(P, Q)$ [6].

We use the resultant or a similar technique to reduce the epipole search to an equation of following type: Let the epipole be $(x, y, 1)^T$,

$$(A_0 + xA_1 + \cdots x^k A_k)(1, y, y^2, \cdots, y^n)^T = 0 \ . \tag{2}$$

We use the determinant of the matrix to determine x, and the right null vector to determine y. Details are presented in the following sections.

3 Epipolar Geometry and the Fundamental Matrix

We first define the problem of epipolar geometry estimation in Section 3.1. Then, in Section 3.2, we reduce the search space to a hemisphere of the epipole. At the same time, we compare the proposed method and the linear 8-point method. In Section 3.3, a simpler formulation of the epipole search is established, which is then reduced to a single-variable equation in Sections 3.4 and 3.5. Finally, a specific method by which to solve the equation is discussed in Section 3.6.

3.1 Basic Formulation

The scene contains two cameras and P feature points. Let $\boldsymbol{x}_p$ be the 3D coordinates of the p'th feature point, and let $\boldsymbol{l}_p$ and $\boldsymbol{r}_p$ be the homogeneous coordinates of the images of the p'th feature point in the first and second images, respectively.

Then, the following relation, known as the epipolar constraint, holds [1]:

$$\boldsymbol{l}_p^T F \boldsymbol{r}_p = 0 \tag{3}$$

where F is referred to as a *fundamental matrix* and conveys the internal and external parameters of the two cameras. The scale of F has no meaning and F should be of rank 2. Therefore, F has seven degrees of freedom.

Our goal is to estimate F from a given set of feature correspondences $\{(\boldsymbol{l}_p, \boldsymbol{r}_p)\}$, by minimizing the *Algebraic* cost function defined as

$$E(F) := \sum_{p=0}^{P-1} |\boldsymbol{l}_p^T F \boldsymbol{r}_p|^2 \qquad \text{or} \qquad E(\boldsymbol{f}) = \boldsymbol{f}^T H \boldsymbol{f} \ , \tag{4}$$

where $\boldsymbol{f}$ is a 9-vector containing all of the elements of the fundamental matrix F in row major order, i.e., $\boldsymbol{f}^T := [\ \boldsymbol{a}^T\ \boldsymbol{b}^T\ \boldsymbol{c}^T\ ]$, where $F^T = [\ \boldsymbol{a}\ \boldsymbol{b}\ \boldsymbol{c}\ ]$, and H is a symmetric matrix defined so that $E(F) = E(\boldsymbol{f})$.

For removing scale ambiguity, the search space of F should be limited so that its Frobenius norm is 1, written hereinafter as $|F|_F = 1$, or $\boldsymbol{f}^T\boldsymbol{f} = 1$.

Note that the algebraic cost function can be regarded as an approximation of the *Sampson error*, defined as

$$E_S(F) := \sum_p \frac{\left(\boldsymbol{l}_p^T F \boldsymbol{r}_p\right)^2}{|F\boldsymbol{r}_p|_C^2 + |F^T\boldsymbol{l}_p|_C^2} \quad \text{where} \quad |(a,b,c)^T|_C^2 := a^2 + b^2 \ . \tag{5}$$

3.2 Simple Equation for λ and $\boldsymbol{e}$

Minimization of the bilinear form $\boldsymbol{f}^T H \boldsymbol{f}$ subject to the constraint $\boldsymbol{f}^T\boldsymbol{f} = 1$ is reduced to the following eigenequation:

$$(H - \lambda I)\boldsymbol{f} = 0 \ . \tag{6}$$

This is the main part of the linear 8-point algorithm. It is easy to show that λ coincides with the residual to be minimized ($\lambda = \boldsymbol{f}^T H \boldsymbol{f}$) and satisfies the 9th order equation $\det(H - \lambda I) = 0$. In this calculation, the rank-2 constraint could not be imposed.

On the other hand, once we know the coordinates of the epipole $\boldsymbol{e} = (x, y, z)^T$, we can easily obtain the rank-2 fundamental matrix, which minimizes the algebraic cost function as follows [1]: Let us define

$$B := \begin{bmatrix} \boldsymbol{e} & & \\ & \boldsymbol{e} & \\ & & \boldsymbol{e} \end{bmatrix}, \ D := \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & & & & \\ & & \boldsymbol{u}_1 & \boldsymbol{u}_2 & & \\ & & & & \boldsymbol{u}_1 & \boldsymbol{u}_2 \end{bmatrix}, \ \text{and } \boldsymbol{g} := D^T\boldsymbol{f} \tag{7}$$

where $(\boldsymbol{e}, \boldsymbol{u}_1, \boldsymbol{u}_2)$ is an orthonormal basis. Obviously, $[B|D]$ is an orthogonal matrix, and $[B|D]$ divides the search space for $\boldsymbol{f}$ into 3D and 6D subspaces. Since the rank-2 constraint requires that $B^T\boldsymbol{f} = 0$, $\boldsymbol{f}$ must lie in the 6D subspace spanned by the column vectors of D. Therefore, the algebraic cost function is $\boldsymbol{g}^T D^T H D \boldsymbol{g}$ and the norm constraint is $\boldsymbol{g}^T\boldsymbol{g} = 1$. Then, $\boldsymbol{g}$ satisfies the following equation:

$$(D^T H D - \lambda I)\boldsymbol{g} = 0 \tag{8}$$

$$\text{where} \quad \det(D^T H D - \lambda I) = 0 \ . \tag{9}$$

From the above equation, $\boldsymbol{f}$ is obtained as $D\boldsymbol{g}$, where $\boldsymbol{g}$ is the least significant eigenvector of $D^T H D$. The resulting matrix must be of rank 2. Compare this equation with eq. (6).

It is important, in the proposed approach, that λ is the algebraic error to be minimized and satisfies the 6th order equation eq. (9), and the coefficients of the equation are a function of the epipole coordinates. Figure 2 (a) shows the minimum one of six solutions of eq. (9), i.e. λ, plotted with respect to $\boldsymbol{e}$, for the image pair shown in Fig. 1. In principle, our goal is now to choose the epipole that minimizes the minimum solution of eq. (9). In Fig. 2 (a), such an epipole is indicated as a small circle, along with other local minima or maxima or saddle points. In the following subsection, we derive a simpler form of eq. (9), in which the coefficients are 6th order polynomials of the coordinates of the epipole.

3.3 Four-Dimensional Polynomial Equation for Epipole Search

Here, we must minimize eq. (4) subject to two constraints, $|F|_F^2 = 1$ and $\det(F) = 0$, or equivalently $\boldsymbol{f}^T\boldsymbol{f} - 1 = 0$ and $B^T\boldsymbol{f} = 0$. The norm constraint is incorporated into eq. (4) by the Lagrange multiplier method as usual, while the rank-2 constraint is incorporated by the penalty method:

$$E(\boldsymbol{f}, \boldsymbol{e}) = \boldsymbol{f}^T H \boldsymbol{f} + \nu \boldsymbol{f}^T B B^T \boldsymbol{f} - \lambda(\boldsymbol{f}^T\boldsymbol{f} - 1) \tag{10}$$

where the penalty weight ν goes to ∞, and λ is a Lagrange multiplier. Note that the last two terms added here are zeroes for the true solution.

As is usual for minimization problems, we start with differentiation:

$$(1/2)\nabla E \;=\; (H + \nu B B^T - \lambda I)\boldsymbol{f} = 0\ . \tag{11}$$

Multiplying $\boldsymbol{f}^T$ from the left to eq. (11), we have $\boldsymbol{f}^T(H + \nu BB^T - \lambda I)\boldsymbol{f} = 0$, and comparing this identity with eq. (10), we can see that $E = \lambda$. This means that the Lagrange multiplier agrees with the cost function to be minimized.

From eq. (11), the determinant of $(H + \nu BB^T - \lambda I)$ should be zero. Although it may involve a considerable amount of symbolic manipulation, it is straightforward to expand the determinant to show the several properties described below. This is a 9th order equation in λ and a cubic equation in ν. In addition, the coefficient for the ν^3 term is a 6th order polynomial in λ, which is hereinafter referred to as f_6. As $\nu \to \infty$, six out of nine solutions of $\det(H + \nu BB^T - \lambda I) = 0$ approach the solutions of $f_6 = 0$. The explicit expression of f_6 is given by

$$f_6(\lambda; \boldsymbol{e}) := \lim_{\nu \to \infty} \nu^{-3} \left| H - \lambda I + \nu BB^T \right| = c_0 + c_1\lambda + c_2\lambda^2 + \cdots + c_6\lambda^6 = 0 \tag{12}$$

The algebraic residual λ should satisfy this equation, which is a simpler form of eq. (9). The coefficients c_i's are 6th order homogeneous polynomials in the coordinates of the epipole. Writing the epipole $\boldsymbol{e}$ as $(x, y, z)^T$, we have

$$c_i = c_{i600}x^6 + c_{i510}x^5y + \cdots + c_{i006}z^6. \tag{13}$$

3.4 High Order Polynomial Equation for Epipole Search

Next, we introduce the 1st and 2nd order derivatives of c_i's, i.e., $\boldsymbol{g}_i := \nabla c_i =: (g_{i0}, g_{i1}, g_{i2})^T$, and $H_i := \nabla\nabla^T c_i$. Note that the homogeneity suggests that $\boldsymbol{e}^T\boldsymbol{g}_i = 6c_i$, and $H_i\boldsymbol{e} = 5\boldsymbol{g}_i$.

In order to find $\boldsymbol{e} = (x, y, z)^T$ that minimizes the residual λ, we use the property whereby the gradient of eq. (12) satisfies:

$$(\ g_0\ g_1\ g_2\ )^T \;:=\; \boldsymbol{g}_0 + \boldsymbol{g}_1\lambda + \boldsymbol{g}_2\lambda^2 + \cdots + \boldsymbol{g}_6\lambda^6 = 0\ . \tag{14}$$

Note that the right-hand side of the equation is intuitively $\omega\boldsymbol{e}$, where ω is a Lagrange multiplier, but considering $\boldsymbol{e}^T(\sum_i \boldsymbol{g}_i\lambda^i - \omega\boldsymbol{e}) = 0$, we obtain $\omega = 6\sum_i c_i\lambda^i = 0$.

We can remove λ by constructing the following d_i's:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \end{bmatrix} := \frac{1}{|\boldsymbol{e}|^{12}} \begin{bmatrix} \mathrm{Res}(g_1, g_2) \\ \mathrm{Res}(g_2, g_0) \\ \mathrm{Res}(g_0, g_1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{15}$$

$$\text{where} \qquad \mathrm{Res}(g_i, g_j) = \det \begin{bmatrix} g_{i0} & g_{i1} & g_{i2} & \cdots & g_{i6} & & & & \\ g_{j0} & g_{j1} & g_{j2} & \cdots & g_{j6} & & & & \\ & g_{i0} & g_{i1} & g_{i2} & \cdots & g_{i6} & & & \\ & g_{j0} & g_{j1} & g_{j2} & \cdots & g_{j6} & & & \\ & & \ddots & \ddots & \ddots & & \ddots & & \\ & & & g_{i0} & g_{i1} & g_{i2} & \cdots & g_{i6} \\ & & & g_{j0} & g_{j1} & g_{j2} & \cdots & g_{j6} \end{bmatrix}, \tag{16}$$

and $g_i = \sum_{j=0}^{6} g_{ij} \lambda^j$. In this case, each of the resultants is a 60th order homogeneous polynomial in x, y and z, and a multiple of $|\boldsymbol{e}|^{12}$. Therefore, we need to consider 48th order quotients only.

We can solve this system of equations in six different methods. In the following, we describe one method, and the other methods are obtained by exchanging the role of variables, say by replacing (x, y, z) with (x, z, y), (y, x, z), (y, z, x), (z, x, y), and (z, y, x), respectively.

Letting $z = 1$ and regarding d_i's as polynomials in y, we have $d_i = \sum_{j=0}^{48} d_{ij} y^j$, where coefficients d_{ij}'s are polynomials in x. Then, the following equation holds for the true x and y:

$$A \begin{bmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^{71} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \text{where } A := \begin{bmatrix} d_{00} & d_{01} & d_{02} & \cdots & d_{0,48} & & & & \\ d_{10} & d_{11} & d_{12} & \cdots & d_{1,48} & & & & \\ d_{20} & d_{21} & d_{22} & \cdots & d_{2,48} & & & & \\ & d_{00} & d_{01} & d_{02} & \cdots & d_{0,48} & & & \\ & d_{10} & d_{11} & d_{12} & \cdots & d_{1,48} & & & \\ & d_{20} & d_{21} & d_{22} & \cdots & d_{2,48} & & & \\ & & \ddots & \ddots & \ddots & & \ddots & & \\ & & & d_{00} & d_{01} & d_{02} & \cdots & d_{0,48} \\ & & & d_{10} & d_{11} & d_{12} & \cdots & d_{1,48} \\ & & & d_{20} & d_{21} & d_{22} & \cdots & d_{2,48} \end{bmatrix}. \tag{17}$$

Therefore, the determinant of A should be 0 for the true x. This condition is a 1,728th order equation in x.

We have obtained a high order polynomial equation for epipole search. However, it does not appear to be feasible to implement this method because the usual double-precision arithmetic is not sufficiently accurate for solving this equation, and the required amount of calculation is enormous. Therefore, in the following subsection, we discuss an approximation.

Moreover, we do not believe this to be the best reduction of eq. (14), which could probably be reduced to a much lower order polynomial by using one of the techniques for calculating the multipolynomial resultant [6]. This is left for a future study.

3.5 Feasible Solution

We need an approximation to construct a feasible equation. We can assume that λ is small enough to neglect the $O(\lambda^2)$ terms, because λ is the residual to be minimized and the typical magnitude of λ is 10^{-2}. When such higher order terms are neglected, $\lambda = -c_0/c_1$. An example is shown in Fig. 2(b), and Figs. 2(a) and 2(b) are almost indistinguishable.

Correspondingly we truncate eq. (14) as follows:

$$\boldsymbol{g}_0 + \boldsymbol{g}_1\lambda = 0 \ . \tag{18}$$

Similar to eq. (15), λ is removed from eq. (18) by

$$(\ d_0\ d_1\ d_2\ )^T := \boldsymbol{g}_0 \times \boldsymbol{g}_1 = 0 \tag{19}$$

where d_i is a 10th order homogeneous polynomial in three variables.

We can solve this equation in a similar manner to that described in the previous subsection. For each selection of six roles of variables, we construct a matrix corresponding to A in eq. (17), which is now of size 15×15, and the determinant is a 75th order polynomial. Figure 3 shows an example of the determinant plotted with respect to x. The polynomial has only a few zeroes, although the order of the polynomial is 75. Figure 4 shows an example of eq. (19), where we can see each of the small circles, representing a minimum, a maximum or a saddle point of λ, lie on the curves where $d_i = 0$ for every i.

3.6 Solving the Equation

We must first calculate c_{ijkl}'s. However, the direct calculation of eq. (12) involves extensive calculation. Therefore, we instead use eq. (9) to obtain the solutions λ_i's for at least 28 linearly independent (x, y, z)'s. Then, c_i's are calculated up to scale [1], from

$$\prod_{i=0}^{6}(\lambda - \lambda_i) = \frac{1}{c_6}\sum_{i=0}^{6} c_i\lambda^i \ , \quad \text{or} \quad \left\{ \begin{aligned} -c_0/c_6 &= \lambda_0 + \lambda_1 + \cdots + \lambda_5 \\ c_1/c_6 &= \lambda_0\lambda_1 + \lambda_0\lambda_2 + \cdots + \lambda_4\lambda_5 \\ &\ \vdots \end{aligned} \right. \tag{20}$$

for each (x, y, z). From these c_i's, an equation system (a least-squares equation system) for c_{ijkl}'s can be constructed, by stacking eq. (13) for more than 28 tuples of (x, y, z)'s.

Then, we easily obtain g_{ij}'s by symbolic differentiation of c_i's.

For searching x, which makes the determinant of A in eq. (17), or its variant of size 15×15, become zero, the search space can be limited within $[-1 : 1]$ because every point (x, y, z) on the hemisphere is covered either twice or three times by the six permutations of the three variables. Then, the range $[-1 : 1]$ is divided into dozens of parts by equally spaced nodes, and the determinant is calculated for each node. If the sign of the determinant changes between two consecutive nodes, there must be a solution in the interval between the nodes, and the root is calculated by the Secant method. We do not calculate the coefficients of the high order polynomial. For each x that makes A singular, the right null vector of A

[1] The scale is not needed. However, if required, we can use the fact that $c_6 = |\boldsymbol{e}|^6$, which can be derived directly from eq. (12), to determine the scale.

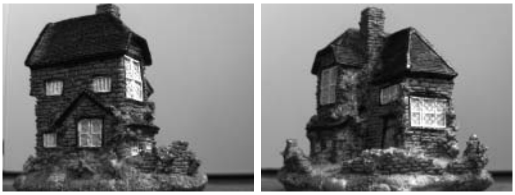
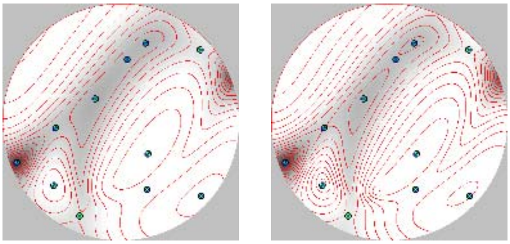


Fig. 1. Example of an image pair



(a) Exact λ (b) Approximation $-c_0/c_1$

Fig. 2. Fish-eye projection of the residual λ. The regions where λ approaches 0 appear as dark regions. Contour lines are also shown.

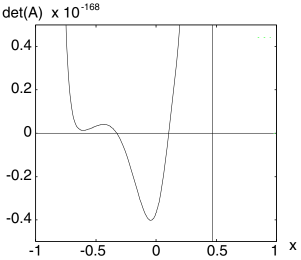


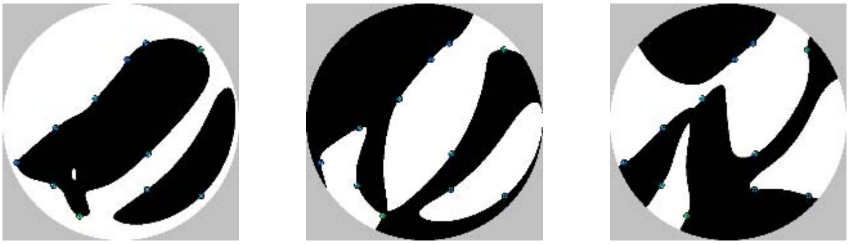
Fig. 3. Determinant with respect to x when $z = 1$. The determinant is a 75th order function of x, but has only three solutions in the range.

contains a geometric series of y, and we can extract y from the series. Otherwise, the solution x should be discarded.

We must also be aware of false solutions, because the equation is satisfied by not only a point that minimizes λ, but also maxima or saddle points. To check this, we use the Hessian $\sum_i H_i \lambda^i$ at the point, at which all of the eigenvalues should be non-negative at a minimum. Once the epipole is obtained, the fundamental matrix is easily calculated, as described in Section 3.2. The required time for the entire calculation is approximately 100 ms on a 3.2-GHz Pentium4 processor.

4 Experiment

In this section, we show the experimental results for a real image sequence of a toy house. Two out of 37 images are shown in Fig. 1. We tracked 100 feature points manually. The average tracking error was approximately three pixels. There are 397 image pairs that have seven or more point correspondences. Thirty images were captured with the camera traveling on a circular trajectory around the object, and the focal lengths are almost fixed. Seven additional images were captured from positions that were not on the trajectory. The distance to the object and the focal length for these additional images may vary. This increases the variation of the epipole position. Figure 5 shows the distribution of the



(a) First element (d_0) (b) Second element (d_1) (c) Third element (d_2)

Fig. 4. The sign of each element of $\boldsymbol{g}_0 \times \boldsymbol{g}_1 = (d_0, d_1, d_2)^T$ is indicated by the black and white areas, and the boundaries between the areas indicate the locations at which $d_i = 0$. The intersections are also indicated by small circles.

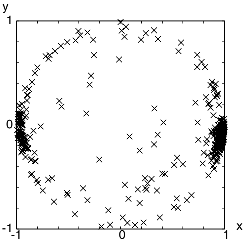


Fig. 5. Distribution of the epipoles for 397 image pairs

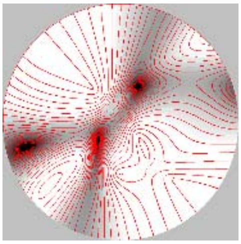
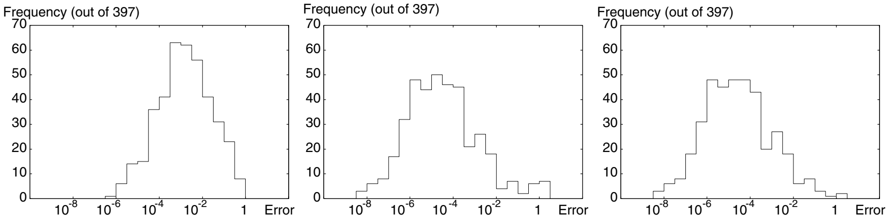


Fig. 6. Example of a λ plot with respect to $\boldsymbol{e}$

epipoles for 397 image pairs. Although many of the epipoles are near $(\pm 1, 0, 0)$, there are image pairs for which the epipoles are near the image center.

Number of Solutions: One of the advantages of the proposed method is that it can detect all of the local minima, whereas the linear 8-point method can generate only one result. The number of epipole candidates, or local minima, varied from 1 to 5. There were 179 image pairs for which the proposed method found only one solution, whereas for the other pairs the proposed method found more than one candidate solutions. Figure 6 shows an example of λ variations in the search space, where there are three local minima. There were an additional 106 such pairs, which indicates that this is not a rare case. For many cases, a smaller number of correspondences causes the search space to have more than one local minimum. We can choose the most appropriate minimum from among these candidates. The choice can be made after a nonlinear refinement.

Accuracy: We compared the proposed method to Sampson error minimization, and the result is shown in Fig. 7, along with similar results for other two methods. In the figure, (a) shows the linear 8-point algorithm (L8P), (b) shows the proposed method with first order approximation, and (c) shows the nonlinear minimization of the algebraic error. Given the epipole $\boldsymbol{e}_1$ obtained by a certain method and $\boldsymbol{e}_2$ obtained by the Sampson error minimization, the error between them is defined as $|\boldsymbol{e}_1 - \boldsymbol{e}_2|^2$. When a method produces several candidates, the



(a) Linear 8-point (b) Approximation $-c_0/c_1$ (c) Exact λ

Fig. 7. Histograms of the errors between the solutions of each method and those of the Sampson error minimization

candidate that gives the smallest error is chosen. We then created a histogram of the errors of over 397 image pairs for each of three methods. In the figure, we can easily see the peak of the histogram is around 10^{-3} for L8P, and around 10^{-5} for the other two methods. This indicates that the proposed method is more accurate than L8P, and thus has lower probability to yield a local minimum. Moreover, since the shape of the histograms shown in (b) and (c) are almost the same, the approximation introduced in Section 3.5 is approximately negligible. In other words, the degradation caused by the approximation is much smaller than the degradation caused by neglecting the rank-2 constraint in L8P. Moreover, the typical error of 10^{-5} means that the proposed method is comparable to the Sampson error minimization.

5 Conclusions

We have described herein a novel approach for estimating the epipole from point correspondences in two uncalibrated views. With a slight approximation, the proposed method reduces the search to a set of six 1D searches, each of which is expressed as a 75th order polynomial equation. Within the limitation of our approximation, the proposed method can provide all of the local minima, and the results are comparable to those of Sampson error minimization. However, it is probable that the presented algorithm is not optimal and so should be improved in a future study.

References

1. Hartley, R. I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)
2. Kanatani, K.: Statistical Optimization for Geometric Computation. Elsevier (1996)
3. Zhang, Z.: Determining the Epipolar Geometry and its Uncertainty: A Review, IJCV 27(2), pp.161–198 (1998)
4. Triggs, B., McLauchlan, P. F., Hartley, R., Fitzgibbon, A. W.: Bundle adjustment — A modern synthesis, LNCS 1883 pp.298–375 (2000)
5. Hartley, R. I.: In Defence of the 8-point Algorithm, PAMI 19-6, pp.580–593, (1997)
6. Cox, D.A., Little, J. B., O'Shea, D.B.: Using Algebraic Geometry, Springer, (1998)

Image Technology for Camera-Based Automatic Guided Vehicle

Young-Jae Ryoo

Deparment of Control System Engineering, Mokpo National University,
61 Dorim-ri, Muan-goon, Jeonnam 534-729, Korea
yjryoo@mokpo.ac.kr

Abstract. This paper describes image technology using a neural network system for an automatic guided vehicle to follow a lane. Without complicated image processing from the image of a lane to the vehicle-centered representation of a bird's eye view in conventional studies, the proposed system transfers the input of image information into the output of a steering angle directly. The neural network system replaces the nonlinear relation of image information to a steering angle of vehicle on the real ground. For image information, the vanishing point and vanishing line of lane on a camera image are used. In a straight and curved lane, the driving performances by the proposed technology are measured in simulation and experimental test.

Keywords: Image sensor, neural network, vanishing point, vanishing line.

1 Introduction

There is increasing interest in intelligent transportation system (ITS) in order to drive safely from a starting point to goal. An important component of intelligent automatic guided vehicle is lane following of lateral motion control of a vehicle. Image and video system plays an important role in lane following because of the flexibility and the two-dimensional view.

Systems that integrate both image sensor and automatic guided vehicle together have received a lot of attention[1-6]. Such systems can solve many problems that limit applications in current vehicles. Recently, some researchers have discussed possibilities for the application of intelligent system in automatic guided vehicle[7-10]. The system comprises major modules: an image sensor, a geometric reasoning, and a lateral control. The image sensor acquires camera images, from which certain segments of lane are extracted by means of image processing algorithms. After the segments are transformed from the image coordinate system to the local vehicle one, they are then transferred to the reasoning module, which assigns them a consistent lane interpretation using local geometric supports and temporal continuity constraints. The lane description is sent to the lateral control system, which generates a steering value for vehicle navigation. The system suffers from heavy computation needed to be complete within the given time. For the reasoning module, the computational complexity is governed by camera parameters while the lateral control being dependent on the parameters of the lane and vehicle. Provided the processing time

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1225 – 1233, 2006.

available is enough, this difficulty might be circumvented with a high speed computing system. Unfortunately, in practice, the processing time is limited, which demands more efficient control methodology from a computational standpoint.

In this paper, a fast image technology for an automatic guided vehicle with image sensors is presented, which uses image information to guide itself along lane. In the proposed system, the position and orientation of a vehicle on a lane are assumed to be unknown, but the vanishing point and vanishing line of the lane are given in a video image. The proposed system controls the vehicle so that it can follow the guidelines of lane. The proposed intelligent system transfers the input of image information into the output of steering angle directly, without a complex geometric reasoning from a visual image to a vehicle-centered representation in conventional studies. The neural network system replaces the human driving skill of nonlinear relation between vanishing lines of lane boundary on the camera image and the steering angle of vehicle on the real ground.

2 Image Technology for Automatic Guided Vehicle

2.1 Feature Extraction from Image Sensor

The information of lane boundary on image is obtained from the image processing of a lane scene. As shown in Fig. 1, the geometrical relationship between the vehicle and the lane can be described in image using the following parameters.

1) The vanishing point (*VP*) describes the lateral position of the current vanishing point from the center of a camera image as shown in Fig. 1 (b) and (d). The current vanishing point depends upon the orientation of the vehicle on the lane as shown in Fig. 1 (a) and (c).

2) The vanishing line (VL_L , VL_R) is defined by the slope of the angle between the current vanishing line and the horizontal line as shown in Fig. 1 (b) and (d). The slope is relative to the deviation defined by the lateral distance of the center of the vehicle from the center of the lane.

The camera image contains the vanishing point and vanishing line of the lane. With these data in the camera picture, the current position and orientation of the vehicle on the lane can be uniquely determined by geometric calculations. On the basis of the above method, the following visual control method is introduced: Fig. 1 (d) is the current camera image, and Fig. 1 (b) is the desired camera image, obtained when the vehicle reaches the desired relative position and orientation on the lane. The vision-based control system computes the error signals in terms of the lateral position and slope derived from the vanishing point and vanishing line respectively in the visual image.

A steering angle generated by the proposed system makes that the vanishing point and the vanishing line on the current image coincide with the desired image. The lateral position of the vanishing point and the slope of the vanishing line computed from the linear vanishing line represent the relative position and orientation of the vehicle on the lane. Then the vehicle is required to move its center to the lateral center of the lane and to parallel the lane by controlling its steering angle. It is significant that the vanishing point moves to the desired point in accordance with human's skill of driving.

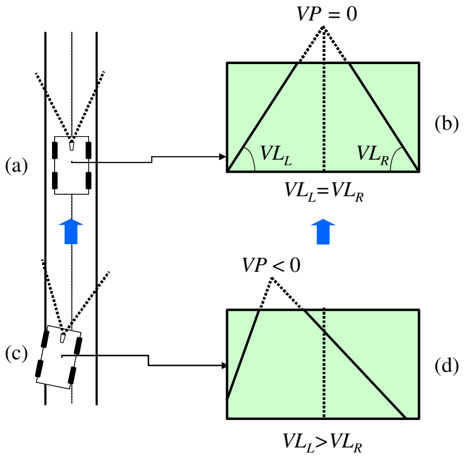


Fig. 1. Relation of camera image of lane view with the orientation and deviation of vehicle on the lane, and the features of vanishing point and vanishing line on camera image

2.2 Using Neural Network

The relation between the steering angle and the vanishing point and vanishing line on the camera image is a highly nonlinear function. Neural network is used for the nonlinear relation because it has the learning capability to map a set of input patterns to a set of output patterns. The inputs of the neural network (x_1 , x_2) are the lateral position of the vanishing point (*VP*) and the slope of the vanishing line (*VL*). The output of the neural network system (y_0) is the steering angle value for the vehicle (δ).

$$\begin{aligned} x_1 &= VP \\ x_2 &= VL = VL_L - VL_R \\ y_0 &= \delta \end{aligned} \tag{1}$$

Learning data could be obtained from human's driving. After the neural network system learns the relation between input patterns and output patterns sufficiently, it makes a model of the relation between the position and orientation of the vehicle, and that of the lane. Thus, a good model of the control task is obtained by learning, without inputting any knowledge about the specific vehicle and lane.

3 Computer Simulations

To simulate in computer as shown in Fig. 2, vehicle model, transformation of coordinate system, and control algorithms should be determined. The used vehicle model is a general model, and has specific parameters. Through transformation of coordinate system, the lane could be displayed on the camera image plane visually.

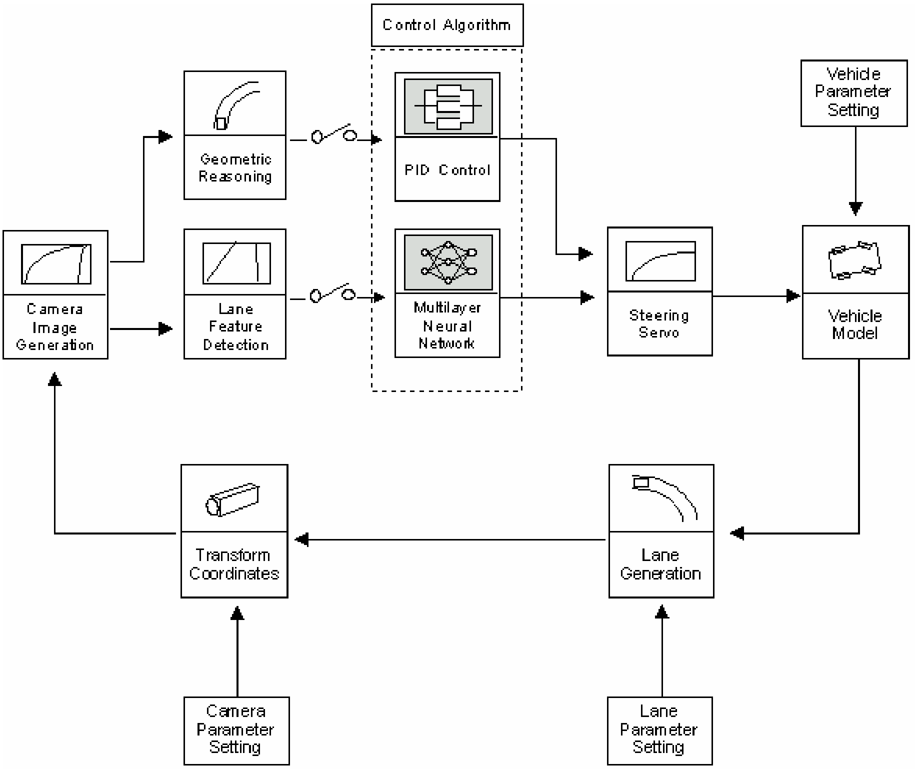


Fig. 2. Structure of simulation

3.1 Vehicle Model

The general model of a vehicle with 4 wheels in the world coordinate system is shown in Fig. 3. The reference point (x_c^W, y_c^W) is located at the center point between the rear wheels. The heading angle θ for X^W-axis of the world coordinate system and the steering angle δ are defined in the vehicle coordinate system, and the model equation is expressed as follows;

$$\begin{aligned} \dot{\theta} &= \frac{1}{L_v} v \sin\delta \\ \dot{x}_c^W &= v\cos\theta\cos\delta \\ \dot{y}_c^W &= v\sin\theta\cos\delta. \end{aligned} \tag{2}$$

Since the vehicle coordinate system is used in control of automatic guided vehicles, the current position (x_c^W, y_c^W) of the vehicle in the world coordinate system is redefined as the point of origin for the vehicle coordinate system. When the vehicle which has the distance L_v between front wheel and rear wheel runs with velocity v, the new position of the vehicle is nonlinearly relative to steering angle δ of front wheel and heading angle θ determined by the vehicle direction and lane direction.

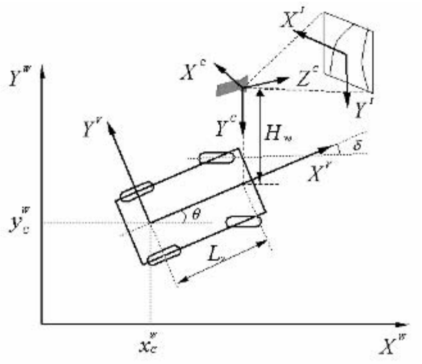


Fig. 3. Vehicle model and camera coordinate transformation

3.2 Transformation from Ground to Image

In order to simulate in computer visually, the lane has to be displayed on the camera image plane. Thus the coordinate transformations along the following steps are needed to determine the lane of visual data from the lane on the world coordinate system :

1) Transformation from the world coordinate system to the vehicle coordinate system. The position (x_c^W, y_c^W) on the world coordinates is redefined as the origin for the vehicle coordinates.

$$\begin{bmatrix} X^V \\ Y^V \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X^W - x_c{}^W \\ Y^W - y_c{}^W \end{bmatrix} \tag{3}$$

where (X^V, Y^V) is a point in the vehicle coordinate system, (X^W, Y^W) is a point in the world coordinate system, (x_c^W, y_c^W) is the reference point located at the center point between the rear wheels, and θ is the heading angle of the vehicle.

2) Transformation from the vehicle coordinate system to the camera coordinate system.

$$\begin{bmatrix} X^I \\ Y^I \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} X^V \\ Y^V \end{bmatrix} + \begin{bmatrix} 0 \\ H_{vc} \\ L_{vc} \end{bmatrix} \tag{4}$$

where (X^C, Y^C, Z^C) is a point in the camera coordinate system, (X^V, Y^V) is a point on the vehicle coordinate system, H_{vc} is a height of the mounted camera from ground, and L_{vc} is the distance of the mounted camera from the center of the vehicle.

3) Transformation from the camera coordinate system to the image coordinate system. (Θ, Φ , Ψ) is the orientation of the camera with respect to the camera coordinates. The image plane (X^I, Y^I) corresponds to the X^C, Y^C plane at distance f (the focal length) from the position of the camera along the Z^C axis. The coordinate transformation is expressed by the rotation matrix R and the focal matrix F.

$$\begin{bmatrix} X^I \\ Y^I \end{bmatrix} = F\left\{ R \begin{bmatrix} X^C \\ Y^C \\ Z^C \end{bmatrix} \right\} \tag{5}$$

where (X_I, Y_I) is a point in the image coordinate system, and (X_C, Y_C, Z_C) is a point in the camera coordinate system.

3.3 Simulation Results

Simulation results for the automatic guided vehicle are shown in this section. The simulation program is developed from programming of vehicle model, transformation of coordinate system, and control algorithms by C++ in IBM-PC.

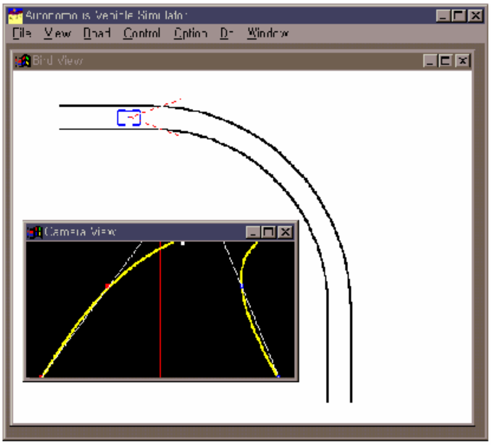

Autonomous Vehicle Simulator
File View Road Control Option D? Window
Bird View
Camera View

Fig. 4. The simulation program shows the camera image view and the bird's eye view

The performance of the proposed visual control system by neural network was evaluated using a vehicle driving on the track with a straight and curved lane as shown in Fig. 5. Fig. 5 shows the bird's eye view of lane map and the trajectories of vehicle's travel.

Fig. 6 shows the vehicle's steering angle during automatic guided driving in the lane of Fig. 5. In Fig. 6, the steering angle is determined from neural network system, and steering actuator model, and vehicle model. The steering angle is almost zero at straight lane (section A), negative at left-turn curved lane (section B), and returns to zero over curved lane (section C). The controller steers the vehicle to left (about -0.14[rad]) at left-turn curve (section B).

A lateral error is defined as distance between the center of lane and the center of the vehicle. As shown in Fig. 7, automatic guided driving is completed in lateral error less than 0.60[m].

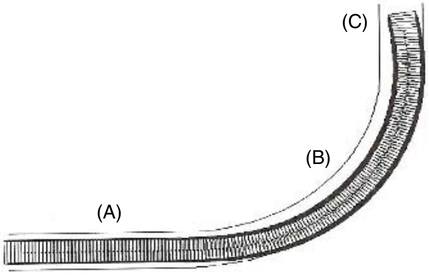


Fig. 5. Trajectories of vehicle driving on curved lane (curvature radius = 6 meters)

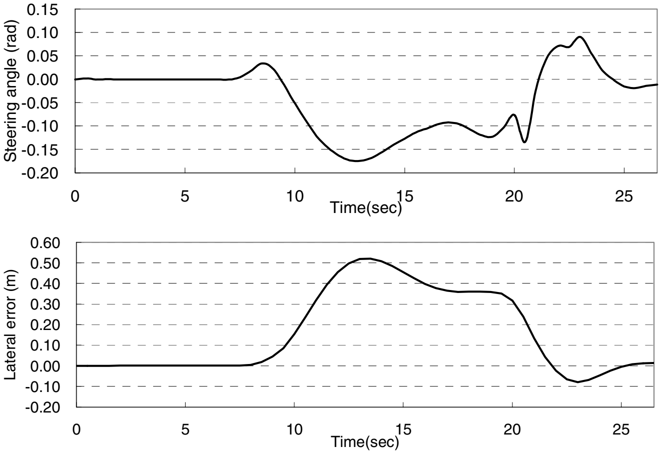


Fig. 6. Lateral error and steering angle

4 Experiments

4.1 Experimental Set-Up

The designed vehicle has 4 wheels, and its size is 1/3 of small passenger car. Driving torque comes from 2 induction motors set up at each rear wheel, and each motor has 200 watts. Front wheel is steered by worm gear with DC motor. Energy source is 4 batteries connected directly, and each battery has 6 volts. And CCD camera is used as a image sensor to get the lane information. The control computer of vehicle has function to manage all systems, recognize the lane direction from input camera image by lane recognition algorithm, and make control signal of steering angle by neural network control. And the control computer manages and controls input information from various signal, also it inspects or watches the system state.

The computer is chosen personal computer for hardware extensibility and software flexibility. Electric system to control composes of vision system, steering control system, and speed control system. Vision system has a camera to acquire lane image and image processing board IVP-150 to detect lane boundary. Steering control system has D/A converter to convert from control value to analog voltage, potentiometer to read the current steering angle, and analog P-controller. Speed control system has 16-bit counter to estimate current speed calculated from encoder pulses, D/A converter to convert command speed from controller to analog voltage, and inverter to drive 3-phase induction motor. Fig. 7 shows the designed vehicle.

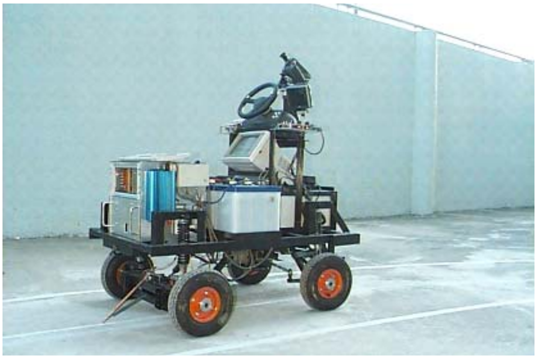


Fig. 7. Designed vehicle

4.2 Automatic Guided Driving Test

The trajectory with a straight and curved lane has the configuration as Fig. 5 evaluated in the simulation study. The lane width is 1.2[m], the thickness of the lane boundary is 0.05[m], and the total length of the lane is set by about 20[m] merged the straight lane and curved lane. The curvature radius of the curved lane is 6[m]. Automatic guided vehicle is confirmed the excellent driving on a straight lane and curved lane as shown Fig. 9.

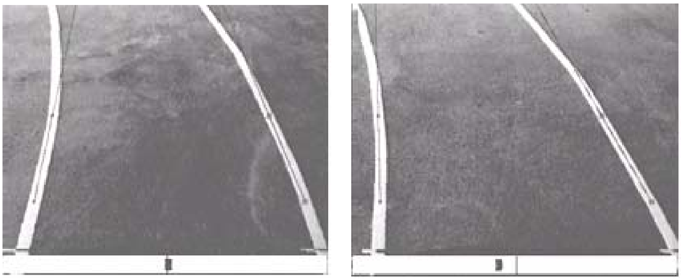


Fig. 8. Camera images, extracted features of vanishing lines and steering angles

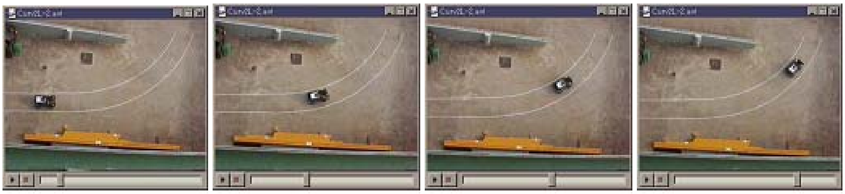


Fig. 9. Automatic guided driving on curved lane

5 Conclusions

In this paper, a scheme for an image technology of an automatic guided vehicle with an image sensor was described using neural network. The nonlinear relation between the image data and the control signals for the steering angle can be learned by neural network. The validity of this image technology was confirmed by computer simulations. This approach is effective because it essentially replaces human's skill of complex geometric calculations and image algorithm with a simple mapping of neural network system.

References

1. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge, (2000)
2. Sadayuki Tsugawa, Vision-based Vehicles in Japan: Machine Vision Systems and Driving Control systems", IEEE Transactions on Industrial Electronics, Vol. 41, No. 4 (1994) 398-405
3. J. Manigel and W. Leonhard, Vehicle Control by Computer Vision", IEEE Transactions on Industrial Electronics, Vol. 39, No. 3 (1992) 181-188
4. C. E. Thorpe, Vision and Navigation, the Canegie Mellon Nablab, Kluwer Academic Publishers (1990)
5. D. Kuan, Autonomous Robotic Vehicle Road Following, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.10, No.5 (1988)
6. A. M. Waxman, J. J. LeMoigne, L. S. Davis, B. Srinivasan, T. R. Kushner, E. Liang and T. Siddalingaiah, A Visual Navigation System for Autonomous Land Vehicles, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 2 (1987) 124-140
7. Young-Jae Ryoo, Vision-based Neuro-fuzzy Control of Autonomous Lane Following Vehicle, Neuro-fuzzy Pattern Recognition, World Scientific (2000)
8. L. Zhaoi and C. E. Thorpe, Stero and Neural Network-based Pedestrian Detection, IEEE Trans. on Intelligent Transportation System, Vol. 1, No. 3 (2000) 148-154
9. Dean A. Pomerleau, Neural Network Perception for Mobile Robot Guidance, Kluwer Academic Publishers (1997)
10. Kevin M. Passino, Intelligent Control for Autonomous Systems, IEEE spectrum, (1995) 55-62.

An Efficient Demosaiced Image Enhancement Method for a Low Cost Single-Chip CMOS Image Sensor

Wonjae Lee[1], Seongjoo Lee[2], and Jaeseok Kim[1,*]

[1] Dept. of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
[2] Dept. of Information and Communication Engineering, Sejong University, Seoul, Korea
ellio@yonsei.ac.kr, seongjoo@sejong.ac.kr, jaekim@yonsei.ac.kr

Abstract. An efficient demosaiced image enhancement method for a low cost single-chip CMOS image sensor (CIS) with an on-chip image signal processor (ISP) is proposed. In a single-chip CIS for handheld devices that requires low power consumption, the ISP implemented must be as small as possible while maintaining high performance. Demosaicing and image enhancement are the largest block in the ISP because they need line memories. In this paper, we propose a new method to minimize the number of line memories. In the proposed method, buffered data for demosaicing were used to make approximated luminance data for image enhancement. This enables line memory sharing between demosaicing and image enhancement. The experimental results showed that the proposed method enhanced the image quality. Thus, both high quality and a small area can be achieved by utilizing the proposed method.

Keywords: Single-chip CMOS image sensor, image signal processor, demosaicing, image enhancement.

1 Introduction

CMOS image sensors (CISs) as image input devices are becoming popular due to the demand for miniaturized, low-power and cost-effective imaging systems such as digital still cameras, PC cameras and video camcorders. CISs have several advantages over CCD (Charge Coupled Device) in terms of opportunity to integrate the sensor with other circuitry. For example, an image signal processor (ISP) can be integrated with a sensor into a single chip to reduce packaging costs.

Several image processing algorithms are used in ISP to improve captured image quality. Among the image processing blocks, demosaicing and image enhancement are the core blocks in ISP. CISs capture only one color channel since the Bayer pattern [1] CFA (Color Filter Array) is used. Therefore, two other colors must be

[*] This work was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) and Yonsei University Institute of TMS Information Technology, a Brain Korea 21 program, Korea.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1234 – 1243, 2006.

interpolated from the neighboring pixel values. This process is called CFA interpolation or demosaicing, and it requires line memories. However, the number of line memories is limited for a low cost single-chip CIS. To acquire high quality demosaiced images, a large number of line memories are essential. In considering the number of line memories for image enhancement, therefore, the new method seeks to minimize the number of line memories while maintaining high performance.

In this paper, we propose a new and efficient demosaiced image enhancement method for ISP integrated CISs. Using the characteristics of conventional high performance demosaicing algorithms, which interpolate the green channel first and then red and blue channels, image enhancement is done without additional line memories.

The paper is organized as follows. In section 2, demosaicing algorithms, image enhancement algorithms, and the conventional architecture of ISP are introduced. The proposed method is presented in section 3. The experimental results are described in section 4, and the conclusions are reached in section 5.

2 Image Signal Processor

Fig. 1 shows a basic structure for an ISP. An image sensor captures the scene. Demosaicing interpolates the missing colors to make color images. After demosaicing, several image processing algorithms are performed to enhance the image quality. Color correction improves color quality. Gamma correction makes the input signal more proportional to the light intensity. The auto white balance removes unrealistic colors to transform the image into what the eyes perceive. The color conversion transforms the color space from RGB to YCbCr, as the YCbCr space is more efficient for image enhancement and compression. Image enhancement improves image contrast or sharpness.

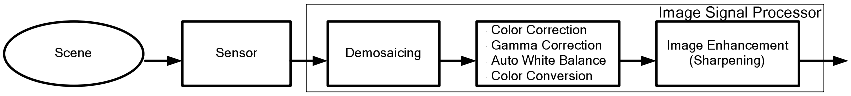
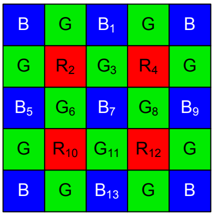


Fig. 1. The basic structure of image signal processor

2.1 Demosaicing Algorithms

CISs can perceive only pixel brightness. Therefore, three separate sensors are needed to completely measure each color channel and make a color image. To reduce the cost and complexity, a single image sensor with CFA is used to capture all three color channels simultaneously. The Bayer pattern, shown in Fig. 2, is widely used as a CFA and provides the array or mosaic of the RGB colors. Since each pixel captures only one color channel, two other colors must be interpolated from neighboring pixel values. If this process is carried out inappropriately, the image suffers from various color artifacts, which results in degraded color resolution.



B	G	B_1	G	B
G	R_2	G_3	R_4	G
B_5	G_6	B_7	G_8	B_9
G	R_{10}	G_{11}	R_{12}	G
B	G	B_{13}	G	B

Fig. 2. Bayer CFA pattern

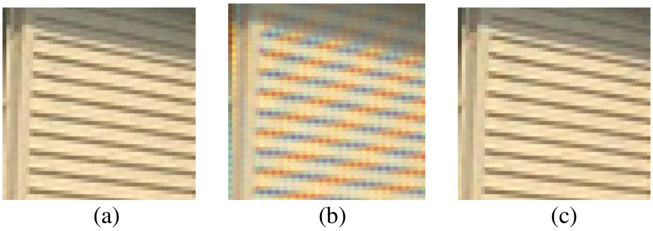
Several demosaicing algorithms have been proposed. Among them, bilinear interpolation is computationally efficient, but it introduces large errors that blur the edges of the resulting image. In adaptive color plane interpolation (ACPI) [2], spatial and spectral correlations are considered to enhance demosaiced image quality. In such images, good quality is maintained even at the edges. It begins by using edge-directed interpolation for the green channel. Referring to Fig. 2, horizontal gradients and vertical gradients are defined as

$$\begin{aligned} DH &= \left|-B_5 + 2B_7 - B_9\right| + \left|G_6 - G_8\right| \ . \\ DV &= \left|-B_1 + 2B_7 - B_{13}\right| + \left|G_3 - G_{11}\right| \ . \end{aligned} \tag{1}$$

Then, the missing green pixel G7 can be estimated as

$$G_7 = \begin{cases} \dfrac{G_6 + G_8}{2} + \dfrac{-B_5 + 2B_7 - B_9}{2} & \text{if } DH < DV \\ \dfrac{G_3 + G_{11}}{2} + \dfrac{-B_1 + 2B_7 - B_{13}}{2} & \text{if } DH > DV \\ \dfrac{G_3 + G_6 + G_8 + G_{11}}{4} + \dfrac{-B_1 - B_5 + 4B_7 - B_9 - B_{13}}{8} & \text{if } DH = DV \end{cases} . \tag{2}$$

Fig. 3 shows the demosaiced image. Many color artifacts are shown in Fig 3(b), which uses bilinear interpolation. However, there are no color artifacts in Fig 3(c), which uses ACPI. In addition, the result image using bilinear interpolation is more blurred than by using ACPI. Therefore, high quality cannot be obtained using bilinear interpolation even though the image enhancement process is applied.



(a) (b) (c)

Fig. 3. The demosaiced image (a) Original image, (b) Bilinear interpolation, and (c) ACPI

2.2 Image Enhancement Algorithms I: Unsharp Masking Filter

Due to the optics and demosaicing, the captured image can be blurred. An unsharp masking filter (UMF) is a well-known technique to improve the visual quality of blurred images by enhancing the edges [3]. It is a simple sharpening operator that derives its name from the fact that it enhances edges via a procedure that subtracts an unsharp, or smoothed, version of an image from the original image. The enhanced image, $\hat{I}_{ij}$, is obtained from the input image, I_{ij}, as

$$\hat{I}_{ij} = I_{ij} + \lambda \cdot (I_{ij} - \bar{I}_{ij}) \ . \tag{3}$$

where λ is a scaling constant and $\bar{I}_{ij}$ is the output of a low pass filter.

2.3 Image Enhancement Algorithms II: Adaptive Contrast Enhancement

When the images captured by the CIS are presented on the display, they may contain low contrast details because of the difference between the display and the scene dynamic range. When scene information with a wide dynamic range is squeezed into the limited range of a typical display, the local details of low contrast are not perceived. To enhance the image quality adaptively, a real time adaptive contrast enhancement method (ACE) was introduced [4]. The local frequency envelope is brought close to the global mean while the high frequency local variations need to be amplified above the contrast sensitivity threshold. This can be modified for implementation purposes by replacing the global mean with the local mean. Fig. 4 shows the block diagram.

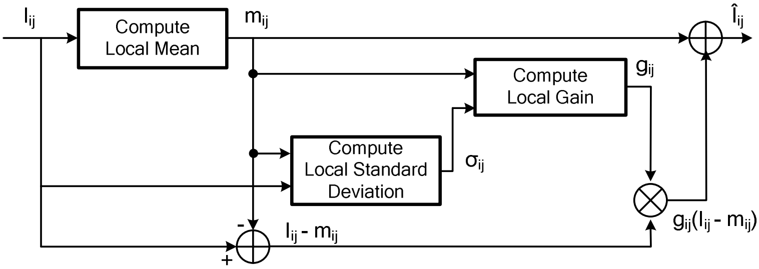


Fig. 4. Block diagram of the adaptive contrast enhancement

The image intensity at each pixel location is transformed based on the local mean (m_{ij}) and local stand deviation (σ_{ij}) estimated on the window surrounding the pixel point. The transformed intensity ($\hat{I}_{ij}$) becomes

$$\hat{I}_{ij} = g_{ij} \cdot (I_{ij} - m_{ij}) + m_{ij} \ . \tag{4}$$

where

$$g_{ij} = \begin{cases} G_{\min} & x < x_1 \\ \frac{x - x_1}{x_2 - x_1} G_{\max} + \frac{x_2 - x}{x_2 - x_1} G_{\min}, & x_1 \le x < x_2 \\ G_{\max} & x \ge x_2 \end{cases} . \tag{5}$$

and x, I_{ij}, G_{min} and G_{max} represent the ratio of m_{ij} to σ_{ij}, the intensity at pixel location (i, j), the minimum value and the maximum value of the variable gain, g_{ij}, respectively. In addition, x_1 and x_2 represent the endpoints of the interval where the variable gain is applied to the difference to amplify the local variations. To prevent the gain from being inordinately large in areas with a large mean and small standard deviation, the local gain is actually controlled by G_{min} and G_{max} in (5).

2.4 Conventional Image Signal Processor

In conventional designs [5-7], a simple bilinear interpolation is used for color demosaicing to minimize the number of line memories. Fig. 5 shows a block diagram of a conventional ISP. The general quality of the images in those conventional designs is not high because bilinear interpolation was used. To enhance image quality, a high performance demosaicing algorithm like ACPI must be used. Even though the demosaiced image quality of ACPI is good, the required line memories are too large for a single-chip CIS when considering the line memories for image enhancement. Therefore, there is a need for a new method to achieve both high quality and low hardware complexity with a small number of line memories.

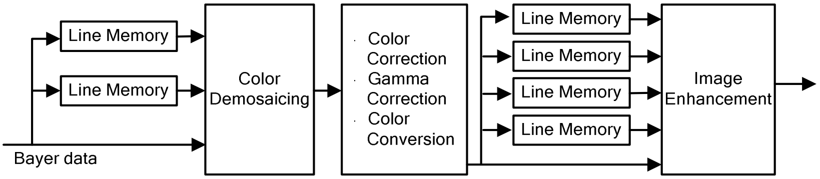


Fig. 5. Block diagram of the conventional ISP

3 Proposed Image Signal Processor

In conventional designs, image enhancement is performed using a luminance channel after color demosaicing. Separated line memories are needed in such conventional architectures. In the proposed method, approximated luminance data are used, enabling line memory sharing between demosaicing and image enhancement. ACPI interpolates the green channel first and then the red and blue channels using the interpolated green channel. This means that all green channel data, which composes about 60% of the luminance signal, are available. Therefore, the proposed method does not need additional line memories for image enhancement.

Fig. 6 shows the demosaicing process of ACPI installed into ISP, which requires four line memories for inputting Bayer data to make the 5x5 window and one line memory for two lines of an interpolated green channel. Since half the green channels are missing for each line, one line memory is enough to store the interpolated green channel. At B19 in the third line of Fig. 6, the green channel is interpolated using the 5x5 window (dash-dot line) and stored in the line memory. Those interpolated and stored green channels are used to make a 3x3 window for red and blue channel interpolation. The missing red and blue channels in the second line are interpolated using the interpolated green channel and the red, green and blue channels stored in the line memory for inputted Bayer data. Data interpolated in the second line become outputs of color demosaicing. In Fig. 6, all the green channel data in the 5x3 window (dotted line) centered at G10 are available after green channel interpolation at B19. Therefore, we can use that green channel data for image enhancement.

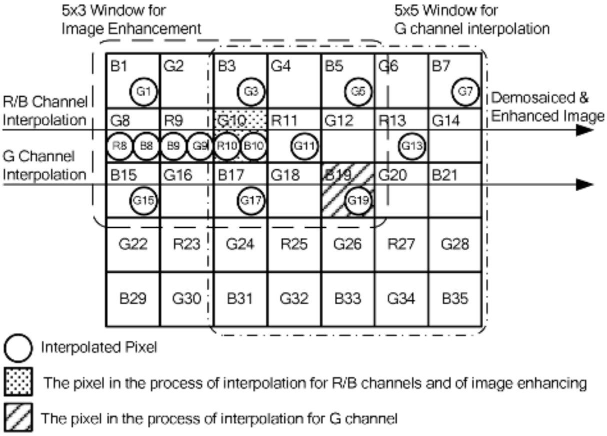

5x3 Window for
Image Enhancement
5x5 Window for
G channel interpolation
R/B Channel
Interpolation
G Channel
Interpolation
Demosaiced &
Enhanced Image
B1 G2 B3 G4 B5 G6 B7
G1 G3 G5 G7
G8 R9 G10 R11 G12 R13 G14
R8 B8 B9 G9 R10 B10 G11 G13
B15 G16 B17 G18 B19 G20 B21
G15 G17 G19
G22 R23 G24 R25 G26 R27 G28
B29 G30 B31 G32 B33 G34 B35
Interpolated Pixel
The pixel in the process of interpolation for R/B channels and of image enhancing
The pixel in the process of interpolation for G channel

Fig. 6. Process of the ACPI

3.1 Proposed Method I: Unsharp Masking Filter

An UMF needs a low-pass filtered image ($\bar{I}_{ij}$). The approximated low-pass filtered image can be calculated as follows using the linearity of the summation.

$$\bar{I}_{ij} = F(I_{ij}) = F(C_1 \cdot R + C_2 \cdot G + C_3 \cdot B) = C_1 \cdot F(R) + C_2 \cdot F(G) + C_3 \cdot F(B)\ . \tag{6}$$

where C_i is the coefficient for color conversion from RGB to YCbCr, and R, G, and B represent red, green and blue, respectively. F(x) is the result of a low pass filter when the input is x. Instead of calculating the exact value of F(x), the approximated value F'(x) can be calculated using the pixels generated during the interpolation process. The interpolated red and blue channels at the second line in Fig. 6 can also be employed for image enhancement using a small buffer.

If we want to enhance G10 in Fig. 6 and use a mean filter (3x3) as a low pass filter, the F'(x) of each channel can be calculated as

$$F'(R) = \left(R8' + R9 + R10' + R11\right)/4 \cdot$$
$$F'(G) = (\sum G_i + \sum G_i')/9 \cdot \tag{7}$$
$$F'(B) = \left(B3 + B9' + B10' + B17\right)/4 \cdot$$

R', G' and B' are the interpolated channel data. The luminance of the current pixel (I_{ij}) can be easily obtained since all data are interpolated and available at the current location.

3.2 Proposed Method II: Adaptive Contrast Enhancement

To adopt the image enhancement method in [4], local mean (m_{ij}) and local stand deviation (σ_{ij}) in (4) and (5) must be calculated. The calculation of the approximated local mean of luminance intensity, m_{ij}, is similar to the process of low pass filtering in the UMF, and the 5x3 window is used.

$$m_{ij} = E(Y) = E(C_1 \cdot R + C_2 \cdot G + C_3 \cdot B) = C_1 \cdot E(R) + C_2 \cdot E(G) + C_3 \cdot E(B) \cdot \tag{8}$$

To obtain the gain, g_{ij}, a local standard deviation has to be calculated. Contrary to the mean, which is a linear equation, the standard deviation is a quadratic equation. Therefore, without all pixel values, it cannot be calculated. In the proposed method, we use the green channel instead of the luminance channel. The ratio of the green channel mean to green channel standard deviation is used to obtain the gain, g_{ij}.

$$g_{ij} = \frac{E_{ij}(G)}{\sigma_{ij}(G)} \cdot \tag{9}$$

3.3 Proposed Image Signal Processor

Fig. 7 shows a block diagram of the proposed ISP. Since a color demosaicing block and an image enhancement block share line memories, it can be implemented using only five line memories, which is the same or less than the line memories used in [5-7].

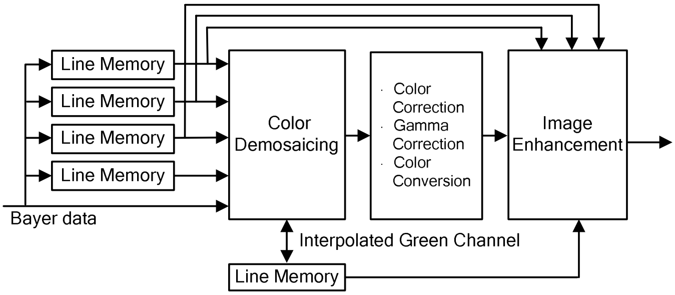


Fig. 7. Block diagram of the proposed ISP

4 Experimental Results

We tested the performance of the proposed algorithm with several natural color images shown in Fig. 8. To show the efficiency of the proposed method, the three methods described below were compared.

1) Method I: Bilinear (3x3 window) interpolation + Image enhancement
2) Method II: ACPI + Image enhancement using luminance data
3) The proposed method: ACPI + Image enhancement using approximated luminance data

Table I shows the performance comparisons. Method II and the proposed method show much better performance than method I, and the number of required line memories for the proposed method is the smallest of the three methods. To show the difference between image enhancement using the luminance channel and the green channel, we calculated the mean squared error (MSE). The averaged MSE between method II and the proposed method is about 4.98 when UMF is applied and about 7.5 when ACE is applied. Even though there are some differences, they are visually the same.

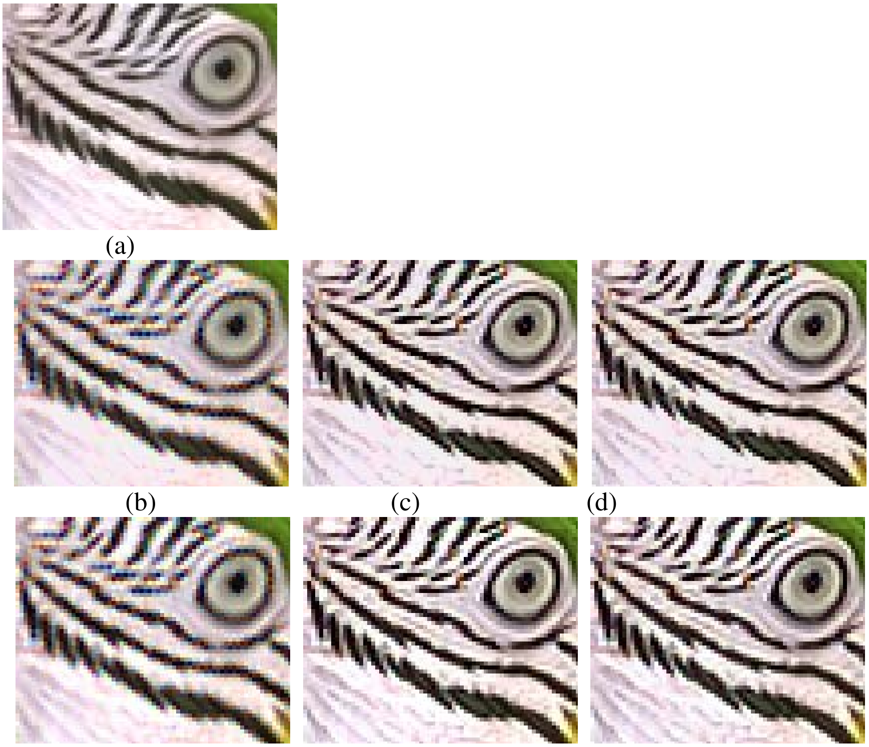
Fig. 8. Color images used in the experiments

Table 1. Performance Comparisons

			Method I	Method II	Proposed Method
Demosaicing Performance	PSNR (dB)	R	28.37	36.45	36.45
		G	32.45	38.25	38.25
		B	28.08	36.27	36.27
MSE	UMF		-	4.98	
	ACE		-	7.5	
# of Line Memory	Demosaicing		2	5	5
	Image Enhancement		4	4	-
	Total		6	9	5

Fig. 9 shows the result images. As shown in Fig. 9, the image quality of the proposed method is almost the same as method II and much better than method I. In Fig. 9(b) and (e), the images are blurred and severe color artifacts are observed.

However, the sharpen images with little color artifacts are shown in Fig. 9(c-d) and (f-g). The experimental results indicate that the proposed method can enhance image quality without using additional line memories.



(a)

(b) (c) (d)

(e) (f) (g)

Fig. 9. Results images: (a) Original image, (b) Method I (UMF), (c) Method II (UMF), (d) Proposed Method (UMF), (e) Method I (ACE), (f) Method II (ACE), (g) Proposed Method (ACE)

5 Conclusion

An efficient demosaiced image enhancement method for a low cost single-chip CIS with an on-chip image signal processor was proposed. For low cost single-chip CISs, the ISP implemented must be small and maintain high quality. In conventional ISPs, demosaicing and image enhancement are completed independently. Thus, they require a large number of line memories. To reduce the number of line memories, simple demosaicing algorithms with low quality are usually used. In this paper, we proposed a new method to minimize the number of line memories. In the proposed method, the buffered data for demosaicing were used to construct approximated luminance data for image enhancement. This enabled line memory sharing between the demosaicing block and image enhancement. To verify the propose method, an unsharp masking filter and adaptive contrast enhancement were used. Experimental results indicated that the proposed approach could enhance image quality without additional line memories. Using the proposed method, other image enhancement techniques can also be applied. Therefore, the proposed method is very useful for the low cost single-chip CIS which requires high performance, low area and low power consumption.

Acknowledgments

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) and Yonsei University Institute of TMS Information Technology, a Brain Korea 21 program, Korea.

References

1. Bayer, Bryce E. “Color imaging array” U.S. Patent 3,971,065
2. J. E. Adams and J. F. Hamilton Jr., “Adaptive color plane interpolation in single sensor color electronic camera” U.S. Patent 5,629,734
3. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods
4. Patrenahalli M. Narendra, and Robert C. Fitch, “Real time Adaptive Contrast enhancement”, IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No. 6, November, 1981, pp. 655-661
5. Yun Ho Jung, Jae Seok Kim, Bong Soo Hur, and Moon Gi Kang, “Design of Real-Time Image Enhancement Preprocessor for CMOS Image sensor”, IEEE Transactions On Consumer Electronics, Vol. 46, No. 1, Feb 2000, pp. 68-75
6. Kim, H., et al.: ‘Digital signal processor with efficient RGB interpolation and histogram accumulation’, IEEE Transactions Consumer Electronics, 1998, 44, pp. 1389–1395
7. Rongzheng Zhou., et al.: ‘System-on-chip for mega-pixel digital camera processor with auto control functions’, ASIC, 2003, Proceedings. 5th International Conference on Vol. 2, 21-24, Oct. 2003, pp. 894 - 897

Robust Background Subtraction for Quick Illumination Changes

Shinji Fukui[1], Yuji Iwahori[2], Hidenori Itoh[3],
Haruki Kawanaka[4], and Robert J. Woodham[5]

[1] Faculty of Education, Aichi University of Education, 1 Hirosawa, Igaya-cho, Kariya
448-8542, Japan
sfukui@auecc.aichi-edu.ac.jp
[2] Faculty of Engineering, Chubu University, 1200 Matsumoto-cho,
Kasugai 487-8501, Japan
iwahori@cs.chubu.ac.jp
[3] Faculty of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku,
Nagoya 466-8555, Japan
itoh@ics.nitech.ac.jp
[4] Faculty of Info. Sci. and Tech., Aichi Prefectural University, 1522-3 Ibaragabasama,
Kumabari, Nagakute-cho, Aichi-Gun 480-1198, Japan
haruki@aichi-pu.ac.jp
[5] Dept. of Computer Science, University of British Columbia, Vancouver, B.C.
Canada V6T 1Z4
woodham@cs.ubc.ca

Abstract. This paper proposes a new method to extract moving objects from a color video sequence. The proposed method is robust to both noise and intensity changes in the observed image. A present background image is estimated by generating conversion tables from the original background image to the present image. Then, the moving object region is extracted by background subtraction. Using color gives more accurate detection than a previous method which used only monochrome data. Color images increase the computational load. The method addresses this problem by using the GPU's throughput. Results are demonstrated with experiments on real data.

1 Introduction

Real-time extraction of moving objects from video sequences is an important topic for various applications of computer vision. Applications include counting the number of cars in traffic, observing traffic patterns, automatic detection of trespassers, video data compression, and analysis of non-rigid motion.

Among the segmentation approaches of moving object, "Background subtraction" is the most basic and speedy approach. However, it works well only when the background image has the constant brightness, and fails when the brightness of the moving object is close to that of the background.

"Peripheral Increment Sign" (PIS)[1] is also proposed for the condition in which the illumination is not constant in a video sequence. It is applicable to

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1244–1253, 2006.

the real-time implementation because the filtering process is simple, but the segmentation of the moving object itself is still liable to be affected by noise.

As for the “Normalized Distance”[2], which was proposed to give a better result even with the effect of illumination change, it is also based on the background subtraction, and is liable to be affected by noise and fails when the degree of brightness is low or when the observed one has the similar texture as the background.

In [3], an approach to estimate the background occluded by a moving object is proposed. It uses the texture and the normalized intensity to the effect of the illumination change, because the texture and the normalized intensity are illumination invariant. This approach assumes the linear change of illumination intensity, and is not applied to the nonlinear intensity changes.

On the other hand, the probabilistic approach using mixture Gaussian model are proposed to remove the shadow region of a moving object from the estimated moving region[4]-[6]. These methods need learning process to extract moving objects and are not applied to the quick illumination changes.

This paper proposes a new method to extract moving objects from a color video sequence. The proposed method is robust both to noise and to intensity changes caused by scene illumination changes or by camera function. A present background image is estimated by generating conversion tables from the original background image to the present image. Then, the moving object region is extracted by a method based on background subtraction. Since the background image, which excludes the moving object itself, is estimated from the observed image and the original background image, the method is applicable to nonlinear intensity changes.

The proposed approach uses a color video sequence and gives more accurate detection than our previous method[7] which used only monochrome data. On the other hand, color sequences increase the processing load, in general. The method addresses this problem by using a Graphics Processing Unit (GPU) throughput. Results are demonstrated on experiments with real data.

2 Proposed Approach

Figure 1 shows the outline of the proposed approach. The proposed approach consists of two processes. The first process estimates the present background image. The second process extracts the moving objects based on background subtraction.

2.1 Generation of Estimated Background Image

Let the RGB color values at (x, y) in the original background image, BG, be denoted by $BG(x, y) = (BG^R(x, y), BG^G(x, y), BG^B(x, y))$. Let those in the observed image at time $t = 1, 2, \dots,$ be denoted by $I_t(x, y) = (I_t^R(x, y), I_t^G(x, y), I_t^B(x, y))$. The region with no moving objects at time t is defined as the background region, A_t.

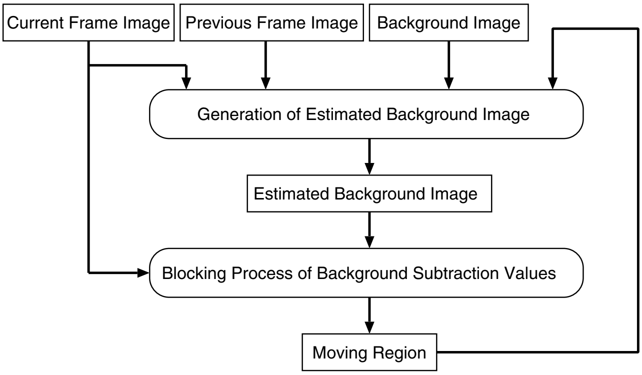

Current Frame Image
Previous Frame Image
Background Image
Generation of Estimated Background Image
Estimated Background Image
Blocking Process of Background Subtraction Values
Moving Region

Fig. 1. Outline of Proposed Approach

When the intensity of the whole image changes, for example owing to the automatic gain control function of the camera, $I_t(x, y)$ in A_t differs from $BG(x, y)$. In this case, moving regions cannot be extracted by simple background subtraction. The background image, BG_t, has to be estimated with high confidence. Then moving regions are extracted using the subtraction of BG_t and I_t.

Conversion tables which convert BG into BG_t are generated to estimate BG_t.

The present background image, BG_t, is estimated through the following procedures:

1. Generate conversion tables
2. Obtain BG_t from conversion tables

Generation of Conversion Tables

Let (x, y) and (x', y') be different points in the image. Suppose, when the overall intensity of the whole image changes, that the value of $I_t^R(x, y)$ is equal to that of $I_t^R(x', y')$ and the value of $BG^R(x, y)$ is equal to that of $BG^R(x', y')$. From this assumption, a conversion table from $BG^R(x, y)$ to $I_t^R(x, y)$ is obtained based on the relation between $BG^R(x, y)$ and $I_t(x, y)^R$ in A_t. Conversion tables for the G and B color channels are obtained in the same way.

Let the candidate for the background region be A'_t. Figure 2 shows how to specify A'_t. First, the absolute value of the differences between I_t and I_{t-1} is obtained. Second, the region with the largest differences is selected by thresholding. Third, A'_t is calculated as the logical AND between the selected region and the last estimated background region A_{t-1}. Here, A_0 is the entire image, which is necessary to estimate BG_1.

When the illumination changes quickly, the approach extracts the whole image region as A'_t. A'_t is treated as a region extracted by dilation process to A_{t-1} in such a case.

The conversion table of each channel from $BG(x, y)$ to $I_t(x, y)$ is made by the RGB values in the obtained region A'_t. These tables convert the RGB color values of BG to the estimated background image BG_t. A histogram shown in

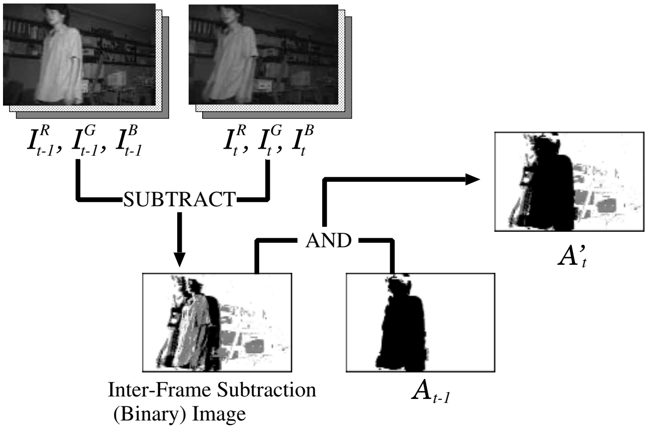

$I^R_{t-1}, I^G_{t-1}, I^B_{t-1}$
I^R_t, I^G_t, I^B_t
SUBTRACT
AND
A'_t
Inter-Frame Subtraction (Binary) Image
A_{t-1}

Fig. 2. Specifying Region A'_t

Figure 3 is produced from each set $(I^R_t(x, y),\ BG^R(x, y))$ in the region A'_t to make the conversion table for R channel.

The conversion table is obtained from this histogram as shown in Figure 4. As shown in the histogram, at the 'value at $BG^R = 26$' in Figure 4, some pixels with the same value of BG^R may not have the same value of I_t under the effect of noise. Therefore, the conversion table uses the median value of a set of pixels at I_t. Since the median value of I_t becomes 11 in this example, the set $(26, 11)$ is obtained and added to the conversion table. The set consists of all pixels which have the same value in $BG(x, y)$ as A'_t. Then, the linear interpolation is applied to the points that do not appear in the conversion table.

Obtaining BG_t from Conversion Tables

The conversion tables convert the RGB color values of BG to the RGB color values of the moving region in I_t. As a result, the background, BG_t, without moving objects is estimated.

2.2 Object Extraction by Background Subtraction

After BG_t is estimated, regions corresponding to moving objects are extracted by background subtraction.

To get a more robust result, I_t is divided into blocks. Each each block is recognized as a background region or not. The total of the absolute values of differences for each block is calculated for each color channel. A block that has one (or more) total below threshold is regarded as a block of an object.

The method changes the threshold value dynamically as the intensity of the whole image changes. The histograms shown in Figure 3 are used to determine the threshold. The procedure for determining the threshold for an R value is as follows: The range with 80% or more of frequency from the mean value is considered. That value is adopted as the threshold for that R value.

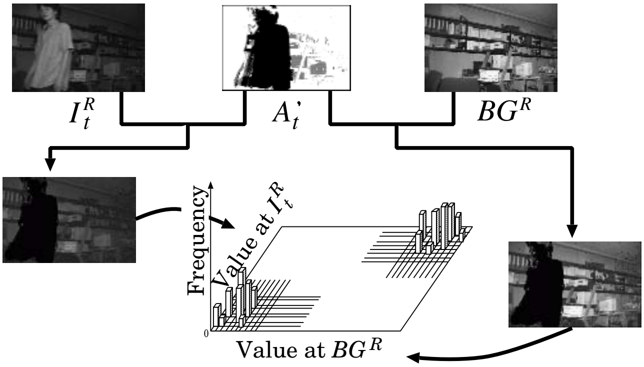


Fig. 3. How to Make Histogram

The threshold for an R value that does not appear in the conversion table can not be determined by the above procedure. In that case, the threshold for the R value is set to be one smaller value than the R value. Thresholds for G and B channels are determined in the same way.

The threshold for each pixel is determined for each color channel. Next, the sum of the absolute value of the differences between $BG_t(x, y)$ and $I_t(x, y)$ and the sum of threshold values for each channel are calculated at each block. Finally, blocks where the sums of the absolute value of the differences become larger than those of the thresholds of all channels are regarded as blocks in the background. Otherwise, they are regarded as blocks in the moving object.

Each block is analyzed by the above process. Subsequently, a more detailed outline of the moving object is extracted. For blocks that overlap the boundary of a moving object, each pixel is classified into either the moving region or the background region, as was the case for the whole block itself.

3 Speed Up Using the GPU

Recently, the performance of the Graphics Processing Unit (GPU) has been improved to display more polygons in shorter time. The geometry and rendering engines in a GPU are highly parallelized. Moreover, the shading process has become programmable via what is called the "Programmable Shader." Current GPUs make geometry and rendering highly programmable.

GPUs are very powerful, specialized processors that can be used to increase the speed not only of standard graphics operations but also of other processes[8]. A GPU is useful for image processing and can be programmed with a high-level language like C. This proposed method uses the GPU for acceleration. While, some approaches using GPU has been proposed recently[9]-[10].

A texture containing a image data should be created to use GPU. The result needed by CPU should be rendered to a texture and CPU should use the texture data. Also, a processing result of a pixel can not be used in a process for other

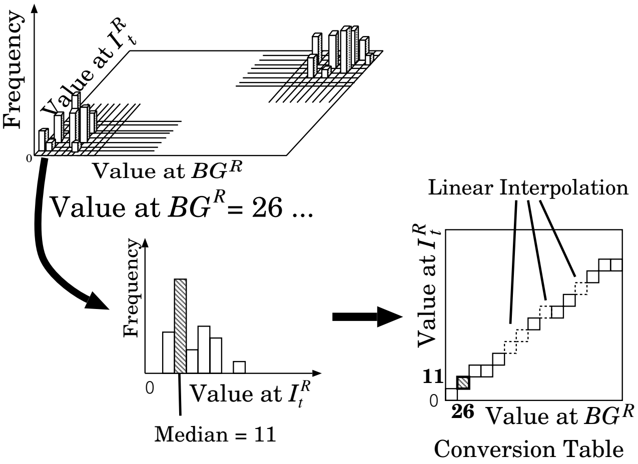


Fig. 4. Generating Conversion Table from Histogram

pixels. So, generating the conversion tables and determining the thresholds are processed at CPU in the proposed method.

The procedure for the proposed method is as follows:

1. Extract A'_t from I_t texture, I_{t-1} texture and A_{t-1} texture, and render the result into A'_t texture
2. Generate the conversion table for each channel from A'_t texture, BG texture and I_t texture, and create texture for the conversion table
3. Determine thresholds for each channel and create texture for thresholds
4. Estimate BG_t from BG texture, I_t texture, A'_t texture and texture for the conversion table
5. Calculate the absolute values of difference between I_t and BG_t, and rendering the result into a texture
6. Determine the threshold for each pixel and render the result into a texture
7. Calculate sum of the absolute value of difference at each block and rendering the result into a texture
8. Calculate sum of the threshold at each block and render the result into a texture
9. Render the result for blocking process into a texture
10. In the border blocks of a moving object, render the result for thresholding process at each pixel into A_t texture. Otherwise, render the result for blocking process into A_t texture

4 Experimental Results

In the experiments, a general purpose digital video camera was used as the input device. A PC with Pentium4 570 and GeForce 780GTX was used to extract

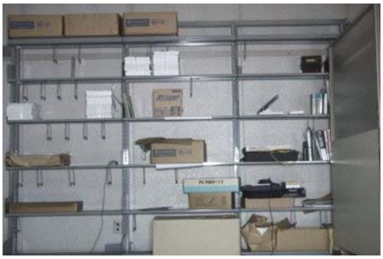


Fig. 5. Background Image Obtained in Advance

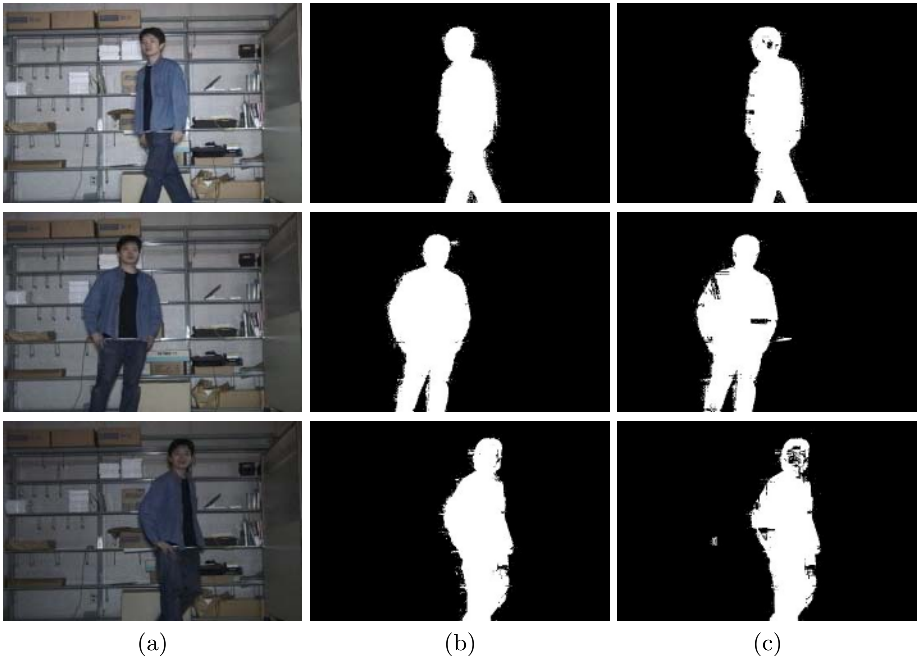


Fig. 6. Observed Images and Results: (a) Observed Images, (b) Results produced by the proposed approach, and (c) Results produced by the previous approach[7]

moving objects. Each target image frame was 720×480 pixels. The corresponding block size was set to 10 × 10 pixels.

First, an indoor scene is used as the original background. The experiment is done under conditions in which the intensity of the light source changes over the course of the image sequence.

Figure 5 shows the background images obtained in advance. Figure 6 shows the observed images and experimental results. Figure 6-(a) shows the observed images, which have different overall brightness from the original background image and Figure 6-(b) shows the experimental results. The white region shows the

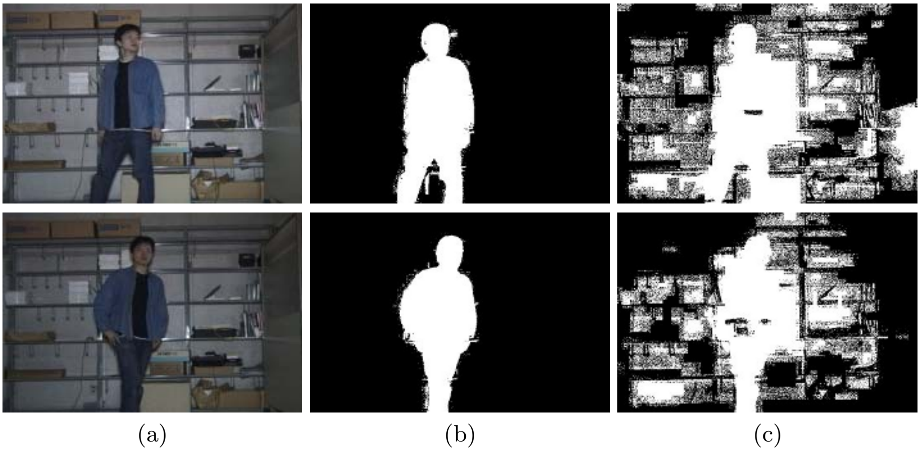


Fig. 7. Results When Illumination Just Changed: (a) Observed Images, (b) Results produced by the proposed approach, and (c) Results produced by the previous approach[7]

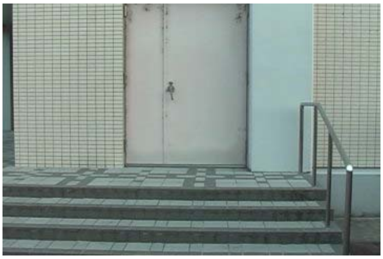


Fig. 8. Background Image Obtained in Advance

detected moving object, and the black region corresponds to the detected background region. Figure 6-(b) demonstrates that this approach can extract the moving object with high performance even under quick illumination changes. Figure 6-(c) shows the experimental results produced by our previous approach [7]. It is shown that the proposed approach works better than the previous approach [7].

Figure 7 shows the results at the moment when the illumination just changed. Figure 7-(a) shows the observed images. Figure 7-(b) shows the experimental results produced by the proposed approach. Figure 7-(c) shows the experimental results produced by our previous approach [7]. The proposed approach has the clear advantage in comparison with the previous approach [7] even when the illumination just changes.

In this experiment, using the GPU performed about 17.9 msec/frame. In contrast, without the GPU performed about 44.7 msec/frame. These results

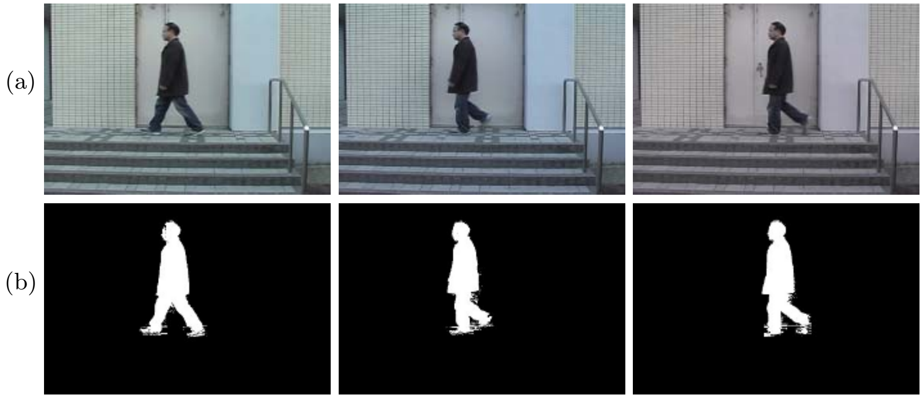


Fig. 9. (a) Observed Images, and (b) Results

show that using the GPU increases processing speed dramatically so that we can extract the moving object in real time.

Next, another experiment was done outdoors, about 20 minutes after sun set.

Figure 8 shows the background images obtained in advance. Figure 9 shows the observed images and experimental results. Figure 9-(a) shows the observed images, which again have different overall brightness from the original background image and Figure 9-(b) shows the experimental results. These results show that this approach can also extract the moving object with high performance even under gradual change in sunlight.

5 Conclusion

A new approach is presented to extract the moving regions in a video sequence. The approach estimates the present background image. It is able to extract a moving object in real time with robustness to quick intensity changes.

Using the GPU results in speed up. GPU can be expected to increase the speed of other image processing operations.

Cases of incorrect detection are remained. Future work includes the detection of shadow regions produced by moving objects.

Acknowledgment

This work is supported by THE HORI INFORMATION SCIENCE PROMOTION FOUNDATION, the Grant-in-Aid for Scientific Research No.16500108, Japanese Ministry of Education, Science, and Culture, and Chubu University grant. Woodham's research is supported by the Canadian Natural Sciences and Engineering Research Council (NSERC).

References

1. Y. Sato, S. Kaneko, and S. Igarashi, "Robust Object Detection and Segmentation by Peripheral Increment Sign Correlation Image," in Trans. of the IEICE, Vol. J84-D-II, No. 12, pp. 2585-2594, Dec, 2001. (in Japanese)
2. Shigaki NAGAYA, Takafumi MIYATAKE, Takehiro FUJITA, Wataru ITO and Hirotada UEDA, "Moving Object Detection by Time-Correlation-Based Background Judgment Method," in Transactions of the Institute of Electronics, Information and Communication Engineers D-II, Vol. J79-D-II, No. 4, pp. 568-576, Apr, 1996. (in Japanese)
3. H. Habe, T. Wada, T. Matsuyama, "A Robust Background Subtraction Method under Varying Illumination" in Tech. Rep. of IPSJ, SIG-CVIM115-3, Mar, 1999. (in Japanese)
4. Stauffer C, Grimson W. E. L.: "Adaptive background mixture models for real-time tracking", in Proceedings of 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). IEEE Comput. Soc. Part Vol. 2, 1999.
5. Stauffer C, Grimson W. E. L. "Learning patterns of activity using real-time tracking", in IEEE Transactions on Pattern Analysis & Machine Intelligence, 2000. 22(8): p. 747-57.
6. P. KaewTraKulPong and R. Bowden: "An improved adaptive background mixture model for real-time tracking with shadow detection" in Proc. of the 2nd European Workshop on Advanced Video-Based Surveillance Systems, Sep. 2001.
7. S. Fukui, T. Ishikawa, Y. Iwahori, H. Itoh: "Extraction of Moving Objects by Estimating Background Brightness" in Trans. of IIEEJ, Vol. 33, No. 3, pp. 350-357, May, 2004.
8. http://www.gpgpu.org/
9. Andreas Griesser, Stefaan De Roeck, Alexander Neubeck and Luc Van Gool: " GPU-Based Foreground-Background Segmentation using an Extended Colinearity Criterion", in Proc. of Vision, Modeling, and Visualization (VMV), Nov, 2005.
10. Ryohei MATSUI, Hajime NAGAHARA, Masahiko YACHIDA: "An acceleration method of omnidirectional image processing using GPU", in Proc. of Meeting on Image Recognition and Understanding (MIRU2005), pp. 1539-1546, Jul, 2005. (in Japanese)

A New Sampling Method of Auto Focus for Voice Coil Motor in Camera Modules

Wei Hsu and Chiou-Shann Fuh

Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C
{r93063, fuh}@csie.ntu.edu.tw

Abstract. We present an effective and efficient sampling table for auto focus searching algorithms to phone camera modules. Many searching algorithms base on focus value curve, and they do not consider the correlation between sampling and image characterization. Proposed table is calibrated by the camera images to meet results of optical lens, sensor response, noise, image pipeline, compressed method, and focus value function. We verify the algorithms in an image signal processor with mega-pixel level cameras. The table reduces total searching steps from 412 to 16, and images are still sharp. Our searching algorithm with the table can focus a scene within 0.6 seconds from infinity to 8 cm. For further progress, they can easily apply to mobile phones.

Keywords: Auto focusing, focus value, phone camera, voice coil motor.

1 Introduction

Auto Focus (AF), Auto Exposure (AE), and Auto White Balance (AWB) algorithms are the most important controls in image quality of digital cameras, and they are key differentiations to other cameras. Image sensors of CCD (Charged-Coupled Device) and CMOS (Complementary Metal-Oxide Semiconductor) are popular today. CCD sensors have better image quality, but prime cost and power consumption are disadvantageous. CMOS sensors with build-in processor are applied to mobile phones and webcam widely, but image quality is just acceptable. A new architecture, ISP (Image Signal Processor) with a raw sensor, has been proposed. Many people believe the architecture can bring a balance between image quality and cost. Actually, a famous company had promoted the design for a time. The sensor provider will not integrate ISP to CMOS sensors of high resolution to reduce the production cost recently.

In a typical ISP, AF finds a focus. AE controls exposure value, and AWB balances color temperatures. Color calibration maps device's colors into standard (or preferred) colors. Image pipeline processes Bayer [1] images to full-color images. JPEG [2] (Joint Photographic Experts Group) encoder compresses images into data streams for communication or storage. In the future, ISP will support H.264 to meet video applications of high compressed rate and quality.

Small dimension and high resolution of phone camera modules are noticeable features. Especially in height of sizes, thickness of mobile phone is a very alluring

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1254 – 1263, 2006.

condition in marketing, but the thickness is usually constrained by camera's height. The height decreases, but optical advantage becomes weakly. In beyond applications, fixed-focus cameras apply to lower resolution (under 2 mega-pixels), and their images are always blurred. An AF function can compensate the weakness, and it effectively upgrades image sharpness, especially at near distance. If high-pixel sensors bond with fixed-focus lens, the output images are still blurred. Therefore, AF is a very important feature in high resolution age.

Normally in AF algorithms, they can be divided into sharpness functions and searching loops. Sharp evaluation and searching servo are related as closely as each in reliability and speed. The computation of sharpness functions is usually greater than searching one. Sharpness functions give statistics to recognize blurred degrees of images. Simple evaluated functions are always influenced by noise. Complex functions can provide more complete information, but their computations are huge against embedded systems. The evaluation of suitable functions is necessary for the purpose.

The optical lens for phone camera has larger depth of filed (DOF), but AF still spends many times to search from infinity to macro distance. Our new sampling table by calibration can reduce searching times and preserve reliability, and proposed searching algorithm performs effectively and efficiently for mobile phones.

2 Focus Value

Focus value is sharp information of images, and auto-focus algorithms measure image quality by focus value functions. Authors present different focus value functions, and we classify functions into gray level, spatial-filter, frequency, and probability domains.

Gray level functions often base on amplitude, energy, gradient, absolute value, and variance [3], [8]. In the spatial-filter domain, high-pass or edge filters evaluate statistics. Tenegrad and Laplacian are famous filter processes [3]. Conveniently, one directional filters in horizontal are developed in real platforms. The unilateral sharp information is incomplete, but it is enough to focus. DCT (Discrete Cosine Transform), FFT (Fast Fourier Transform), and wavelet are used in frequency analyses. Human coding length can be focus value [7] in probability. In JPEG compression, frequency data are encoded by Huffman coding. In-focus images have more complex distribution in frequency lines, and stream length is larger.

Sum-Modulus-Difference (SMD), Laplacina, FFT, or coding length is the representation in each domain. Focus value curves of Fig. 1 show a sequence of JPEG from infinity to macro range by five focus value functions. Gray level amplitude often has many local maximum values, and many searching methods lose. SMD algorithm is nice in the images, because it have a sharp and clear curve. FFT fits to analyze images, and Laplacina is a robust method in many objective conclusions. Images are very complex results from captured devices, and they are consisted of the scenes, noise, sensor response, optical lens, image pipeline, compressed specification, and captured condition. After quantization, JPEG images have lost some information, and it is added some extra noise.

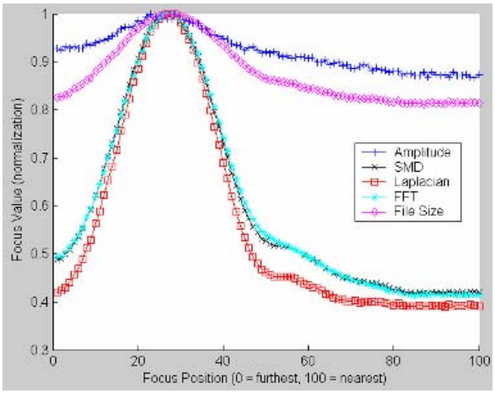


Fig. 1. Focus value curves are by five different functions. Sequential JPEG images are captured from the furthest end to the nearest end step by step. Gray level amplitude, SMD, Laplacian, FFT, and file size algorithms have different image features by the curves.

Different functions perform different image features. Many methods normalize results, and we can ignore the operation in search for computational reduction. For the fastest method, we only read file size as focus value, and it costs nothing in time and space. Frequency domain methods have the most complete information, but too complex. Algorithms in gray level domain are the fastest, but they are sensitive to slight variation of images. Methods in the spatial-filter domain are applicable in embedded systems, but a hardware engine should be implemented for real-time requirements. We can combine, modify or transfer focus value functions to improve statistical reliability [4]. Two or more focus value functions can be computed at the same time, and there is more information for search.

In different applications, we capture images by many data formats. Focus value functions usually measure sharpness by luminance and green color. We also can consider file sizes or stream lengths from compressed data. If there is a DCT hardware engine of an encoder, we can use frequency transformation for the most complete and accurate statistics conveniently. Focus value functions are difficult to say what is the best, but we can choose a suitable one to meet specific requirements under constraints of system ability and computational time.

3 Focus Sampling

AF sampling is scenic capture from different focus distances. A focus value curve records focus values at all of focus positions, and we usually evaluate AF accuracy by the curve. Window segment and adaptive step size have flexible applications to improve AF performance. A sampling table by calibration can speed up focus time, and we discuss with them as follows.

3.1 Focus Value Curve

Stepping motor moves lens by gear rotation, and VCM (Voice Coil Motor) drives lens by coil displacement to change different focusing. Input images of captured devices are discrete signals, and fixed focus positions must be repeated under mechanical and electrical controls. Generally, total steps can cover all focuses. In searching rang, searching algorithms find a focus to decrease unnecessary lens movement. Global search scans all steps, and it records focus value every step in Fig. 2.

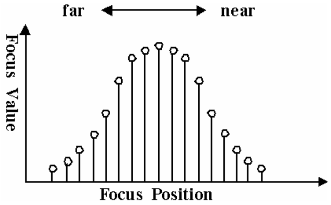


Fig. 2. Discrete signals of a focus value curve. Vertical axis is amplitude of focus value, and focuses at different position are on horizontal axis. Global search scans all steps to plot the curve, and we usually analyze AF methods by the curve.

Noise, camera shaking, and object movement often make vibration in a curve. The situation is always confusing searching algorithms. The maximum statistic has the sharpest image in the most definition of sharpness functions, and focus value of in-focus range is greater than one of out-focus range [6]. After a search, AF moves lens to the position with the maximum statistic to finish the action. Other searching algorithms can be imagined that they jump on a focus value curve by global search to find their best focus.

3.2 Window Segment

The purpose of window segment can reduce computation and improve 3A (AF, AE, and AWB) performance as Fig. 3. Statistics by a full image is an average result, and we are hard to see scenic details. We segment an image into many blocks, and compute their information. Statistics of blocks usually include focus value, color average, or color summation.

A small number of blocks lose details, and majority has huge computation. Basically, bigger image sizes and more blocks are better for precise evaluation. If there is no constraint in computation, the best preciseness is by one pixel per block. The block-average operation can reduce noisy interference, and it also reduces computation to meet the real-time requirement. In our experiments, we propose 80 to 200 blocks in 640 by 480 pixels.

3A algorithms could give different weights to blocks or select areas for precise controls. Central, spot, and matrix of AE metering are typical applications. Many AWB algorithms analyze statistical distribution in specific domains to decide samples for color correction. A central area is important in AF as Fig. 3 (a), and we can give suitable weights to correspond to human vision by location. Relatively, peripheral blocks are unconcerned even though we can ignore them. If we know the subject in

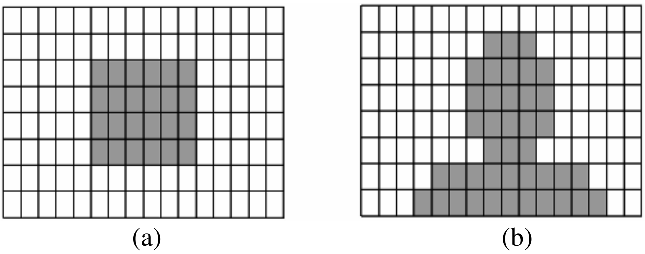


Fig. 3. Window segment. The blocks in dark color are interesting features. We can give weights on the blocks to enhance reliability. (a) A central area. (b) An adaptive area.

which blocks as Fig. 3 (b), AF can refer to the blocks for the most precise search. Such as object detection and face tracking are the applications for adaptive areas. Window segment is very flexible and effective for algorithmic development, and it is tendency towards ISP.

A focus value curve by a small number of blocks makes a steep mountain, and it easily differentiates from a peak or hillside. The disadvantage is sensitive to noise, and a local maximum often appears. Smooth curves are accumulated by more blocks, but searching algorithms decide the peak slow. It is trade-off between computational blocks and focus value reliability. Many AF methods choose different areas between out-focus and in-focus range for accurate and fast purposes, and the areas are different central sizes [4], [5], [6]. In out-focus range, a smooth curve avoid local maximum to converge to in-focus range fast. AF can accurately find a peak in a steep curve of in-focus range through area-size exchange.

3.3 Step Size

Searching algorithms change step sizes to speed up convergence. Global search uses the minimum step to sample all images in searching range. It is the slowest method, but the most accurate. Binary search compares focus value to use decrement of half steps to find the maximum value back and forth. Fibonacci search is similar to binary search, but the step sizes are arranged by Fibonacci sequence. Binary search and Fibonacci search are faster than global search, but their accuracy is doubtful.

Many searching methods consider amplitude [4], slope [5], or ratio [9] of focus value to change step size during hill-climbing. Methods of Adaptive step size can decrease focus time efficiently. Adaptive methods are very difficult to decide step sizes in different searching stages, and they do not concern with scenes, optical lens, and focus value functions. Focus value cures are very different in many cameras, and controlled parameters in the AF must be fined tuning to improve performance and reliability for a specific platform.

3.4 Sampling Table

Searching algorithms become efficiently and effectively by a sampling table. The table records focus steps, and a searching algorithm tracks the table to move lens position. Fig. 4 illustrates non-linear relation between object-distance and focus steps by a phone camera module.

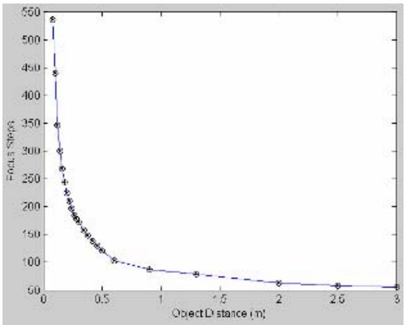


Fig. 4. An object-distance and focus steps curve. In the camera module, the optical effective steps are around 490 steps. DOF is narrow at near end, but it is wide at far distance.

Optical lens has the DOF feature, and we can utilize lesser steps to represent all steps. Lens movement should be involved in the table as a result of non-linear relation for efficient searches. A sampling table can be built by DOF or fixed distances with probability. By fixed distances, the purpose is focus at some fixed distance, but it is discontinuous in optics. By DOF solutions, sampling steps have optical difference, and they can reduce probability of local maximum. Searching algorithms with the table also can avoid noisy interference to improve accuracy. Because of the optical nonlinear relation, lens movement should not consider logic judgement only. We should track specific focuses in sampling to improve AF performance.

In adaptive step controls, a big step could jump a peak, and a small one is slow. Adaptive controls read focus value to adjust steps. Normally, high-contrast scenes have a sharp focus value curve, and low-contrast scenes have a gradual one. Searching algorithms often confuse in-focus with out-focus, because they never know the curvature. Noise also is a major cause to interfere in focus value curves. Preview mode with high frame rate needs huge image data, and it accompanies more noise. If searching algorithms can base on optical continuous features to search, it would not lose necessary sampling steps. An object at the far end with large DOF could be use small steps, and we can use big steps at the near end respectively. It is the reason to keep reliability and improve speed. Searching steps are reduced, and searching times are saved without quality loss. Images are complex. If we only consider the DOF feature, it is insufficient. We present a new sampling method to calibrate by a camera.

4 The Algorithms

Our proposed method has a sampling table and searching algorithm for VCM lens in camera modules. The table establishment is described in section 4.1, and a fast and robust searching method is stated in section 4.2.

4.1 The Sampling Table

Fig. 5 draws a focus value curve, and DOF makes clear images in a range. An acceptable threshold differentiates between blurred and clear image. Clear images have over acceptable threshold in the curve, and blurred images are opposite.

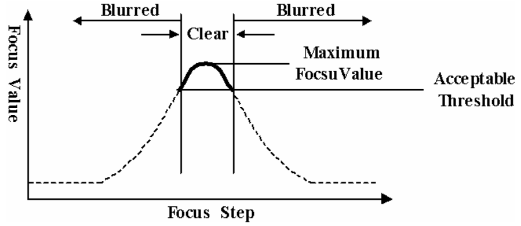


Fig. 5. A focus value curve corresponds to clear and blurred images by an acceptable threshold. Clear images are at the hilltop, and they are in a nusmbers of focus steps.

The curve has the maximum focus value (*MFV*), and the blurred ratio (*BR*) is calculated by

$$BR = Focus\ Value/MFV \tag{1}$$

For an efficient search, we build the sampling table from a blurred end (infinity) to the other blurred end (macro). At the first, we need give an acceptable threshold ratio (*ABR*) to decide clear or blurred images. A testing chart is set at infinity distance, and global search finds the focusing step. Focusing is moved toward near direction step by step until *BR* is smaller than *ABR*. At the condition, we record the focus step into the proposed table, and set the testing chart at the focus step with a new *MFV* by global search. After the iterative processes from the furthest to nearest end, we can build the sampling table. The table by the current focus value function can cover all steps in searching range. Searching algorithms refer to the specific steps which are peaks of curves efficiently in Fig. 6, and all scenes have sharpness over *ABR* by the sampling.

ABR should be fined tuning by a target camera. Focus value curves of a fixed scene are very different by different cameras and focus value functions. Noise and artifact interfere in adjustable degrees of edge enhancement. Therefore, we should estimate the image quality to decide *ABR* to meet the customization.

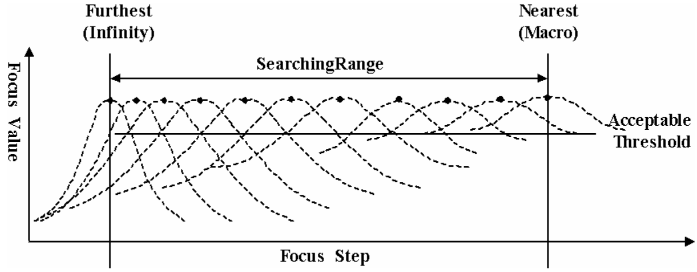


Fig. 6. Sampling steps for auto focus. A focus value curve is calibrated one by one in searching range. Every curve is different gradient after normalization, and the sampling steps are at the peaks.

4.2 Searching Algorithm

Our proposed method has an initial and searching loop stage, and we consider the central window segment for focus value evaluation. Fig. 7 draws the search flow chart.

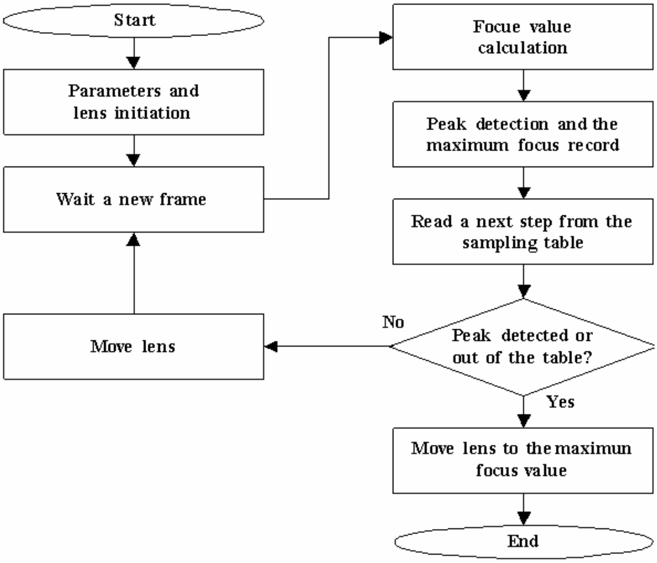

Start
Parameters and lens initiation
Wait a new frame
Move lens
Focue value calculation
Peak detection and the maximum focus record
Read a next step from the sampling table
Peak detected or out of the table?
No
Yes
Move lens to the maximun focus value
End

Fig. 7. Proposed auto focus searching. Initial and searching loop stages are the backbone in our auto focus servo. A new frame drives the loop progress. Peak detection decides a peak of a focus value curve to reduce searching times. Finally, lens is moved to a focusing.

In the initial stage, we move lens to focus infinity and reset controlled parameters. The *basic focus value* is recorded for reference of the current scene. The loop progress refers to the sampling table frame by frame, and focus values renew synchronously. The maximum focus value is also updated every frame.

When the table runs out, it terminates the loop. Peak detection also can break the loop, and it bases on the *basics focus value*. We assume the focus value of in-focus range is over 1.1 times than one of out-focus range [9] in high-contrast scenes. If the focus value decreases on going and accumulates to seven frames under 0.9 times condition of *basics focus value*, lens has been over the peak. If peak detection does not break the loop, all of sampling steps are be scanned. After the loop termination, lens is moved to the position with the maximum focus value, and we finish AF.

5 Experimental Results

In our experiments, we use a VCM camera module which is 10 x 10 x 7.3 mm^3, and the effective searching range is from infinity to 8 cm. The actuator driver outputs the voltage to make lens displacement for VCM. The driver has 10 bits resolution for voltage control, and we can arrange 1024 steps. Effective steps (infinity to 8 cm) are 412 steps in searching range, and others cannot be focused. Popular CMOS sensors of 2 and 3 mega-pixels are verified by our AF algorithm. The sensors can provide 10 bits Bayer data of 640 by 480 pixels at 30 fps (frame per second) in preview mode, but we control in 28.5 fps for better image quality. Our experimental ISP has ability to process the image size in real-time. The 3A hardware engine in ISP calculates red, green, blue, and focus value statistics with 12 by 8 window segment. The computational time and lens displacement can be finished within a frame flush, and the software is developed in language C.

Optical DOF of VCM camera modules is larger, and we reduce 4 steps to 1 step for global search evaluation. Global search spend 3.61 seconds in a round by 103 steps. In our proposed table, the acceptable threshold ratio is set to 0.94 to calibrate our camera module, and we reduce 412 steps to 16 steps. Our method spends 0.56 seconds in the worst case, and the least time is 0.32 seconds. Peak detection can reduce searching steps in high-contrast scenes, and we spend less time for a search.

In accuracy, we base on the sampling table to search in searching range. Peak detection is not work in low-contrast scenes for reliability. If the object is at the near end, the peak detection also looks ineffectively. In other words, our search scans all of steps in the sampling table. The maximum focus value of global search indeed is different from ours, but the differences are within acceptable range. Therefore, images have sharpness over acceptable vision. Actually, it is fast even though we do not care the peak detection. Our AF scans all of focuses within 0.6 seconds in our platform.

6 Conclusion

Our proposed algorithm performs not only fast but also accuracy by camera modules and platform. In the sampling table, focus values of neighbor values are very different, and the sampling method can avoid local maximum situations possibly. The table has the minimum and most efficient sampling steps for other searching algorithms. For more accurate, we also can search the maximum focus value among three largest sampling steps.

Images are consisted of the scenes, sensor response, optical lens, image pipeline, and compressed method. Especially in image pipeline, edge enhancement functions can compensate sharpness, but too strong operation is not naturally. Noise often emphasizes at the same time. In the current stage, we estimate the acceptable blurred ratio by vision. PSNR (Peak Signal-to-Noise Ratio) is a good method to evaluate a difference of two images. When we change focal lengths, the scenes are different. We are hard to evaluate the blurred degree by PSNR. The combination of aperture sizes and optical zooms has different DOF, and tables must be calibrated by conditions.

In the future works, we will focus on the acceptable blurred ratio for objective evaluation. The developed software can be transferred to the family chips easily, and we will apply them to ISP chip for further progress. Zoom tracking has the similar concept, and we will transfer the idea.

References

1. Bayer, B.: Color imaging array. In: U.S. Patent No. 3,971,065. (1976)
2. Wallace K.: The JPEG still picture compression standard. IEEE Trans. on Consumer Electronics.(1991)
3. Chern N. K., Neow P. A. and Jr M. H. A.: Practical issues in pixel-based autofocusing for machine vision. Int. Conf. On Robotics and Automation. (2001) 2791- 2796
4. He J. et al: Modified fast climbing search auto-focus algorithm with adaptive step size searching technique for digital camera. IEEE. Trans. on Consumer Electronics, Vol.49, No.2. (2003) 257-262
5. Choi K. and Ko S.: New autofocusing technique using the frequency selective weighted median filter for video cameras. IEEE Trans. on Consumer Electronics, Vol. 45, No. 3. (1999) 820-827
6. Lee J. S., Jung Y. Y., Kim B. S., and Ko S. J.:An advanced video camera system with robust AF, AE, and AWB control. IEEE Trans. on Consumer Electronics, Vol. 47, No. 3. (2001) 694-699
7. Weintroub J., Aronson M. D., and Cargill E. B.: Method and apparatus for detecting optimum lens focus position. In: U.S. Patent No. 0,117,514. (2003)
8. Lee, J.H et al: Implementation of a passive automatic focusing algorithm for digital still camera. IEEE Trans. on Consumer Electronics, Vol. 41, No. 3. (1995) 449-454
9. Kehtarnavaz N. and Oh H. J.: Development and real-time implementation of a rule-based auto-focus algorithm. Journal of Real-Time Imaging, Vol. 9, Issue 3. (2003) 197-203

A Fast Trapeziums-Based Method for Soft Shadow Volumes

Difei Lu[1] and Xiuzi Ye[1,2]

[1] College of Computer Science, Zhejiang University,China
[2] SolidWorks Corporation,USA

Abstract. One of the best choices of multimedia for fast, high quality shadows is the shadow volume algorithm. However, the calculation of detailed soft shadows is one of the most difficult challenges in computer graphics in the case of area light source. In this paper, we present a new fast trapeziums-based algorithm for rendering soft shadows using a single shadow ray for each shadow pixel. Compared to other soft shadow methods, our algorithm produces very pleasing smooth and artifact-free soft shadow image while executing one order of magnitude faster. Our main contribution is a trapeziums-based method for quickly determining the proportion of area light which overlaps with occluders as seen from the shadow point to be shaded rather than using sample points on the area light source. To speed up calculation a bound box of projected light source is used to relate potential silhouette edges with shadow points beforehand. We demonstrate results for various scenes, showing that detailed soft shadows can be generated at very high speed.

Keywords: shadow algorithms, visibility determination, soft shadow volume.

1 Introduction and Previous Work

Soft shadows are a result of the continuous variation of illumination across a receiving surface when the light source becomes partially occluded by other objects in the scene. Their appearance is mainly controlled by the shape and location of penumbra regions, which are the regions on a receiver where the light source is partially visible. Soft shadows play a key role in the overall realism of computer generated images. In this paper we present an extended soft shadow volume algorithm based on the earlier penumbra wedge-based methods [1, 2, 3, 4] to produce physically quality shadows from planar area light sources.

[5, 6] present excellent surveys of the vast literature on shadow algorithms, and we will only briefly review here some of the main approaches, especially the few that create physically-based soft shadows from area light sources. Using penumbra wedges [7] is fundamentally more efficient than processing the entire occluders. Earlier approximate soft shadow volume methods have been targeting interactive rate performance based on hardware. Ray tracing algorithms compute shadows by casting a ray between a scene point lying and a light source [8]. This method gets great result, but is quite expensive because each ray must in effect sample the scene for potential

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1264 – 1272, 2006.

occluders. [9] parameterizes rays with a spread angle using cones. The technique can approximate soft shadows by tracing a single cone from a surface point to an area light source. [1] describes a new soft shadow volume algorithm that generalizes the earlier penumbra wedge-based methods to produce physically correct shadows from planar area light sources. The algorithm is targeted to production-quality ray tracers, and generates the same image as stochastic ray tracing, but substantially faster.

The fundamental operation in soft shadow generation is determining the visibility between an area light source and a shadow point. The proportion of the visible light source determines the brightness of the shadow point. The algorithm presented in [1] approximates an area light source with a set of light samples and may produce some artifacts on the shadow image. We use trapeziums to present the area light source exactly to get more smooth and artifact-free results. [10] uses a simple extension to ray tracing to create visually approximate shadows with little extra computation. It is interesting and useful but the result is not very pleasing. Our method is based on ray tracing and the result is better than [10].

Extending from [1], we introduce bound rectangles for quickly determining the set of edges whose projections from a shadow point to the plane of an area light overlap with the light source. We don't construct penumbra wedge for finding silhouette edges to avoid creating complex penumbra wedges. In order to speed up calculation we store the potential silhouette edges into a grid data structure. Each cell of the grid represents a shadow point and contains many related potential silhouette edges. This operation is carried out in the beginning of calculation. After finding out all potential silhouette edges for all shadow points, we check all these edges to get exact silhouette edges for every shadow points. Then for a given shadow point P we project its exact silhouette edges to the plane of light source. From these projected silhouette edges we construct many trapeziums to analyze visibility of light source. We will present the details in latter sections.

2 Obtaining Soft Shadows with Ray Tracing

In this section we present the basis of our soft shadow volume creating technique. The pipeline for computing soft shadows is adapted from [1]. There are five steps in the process, i.e., (1)Finding out all global potential silhouette edges from the area light source; (2)Using bound rectangles to get local potential silhouette edges for each shadow point; (3)Find out exact silhouette edges from result of (2) from each shadow point; (4)For a given shadow point, projecting its silhouette edges to the plane of light source and modifying the projected silhouette edges according to the rules detailed later; (4)Splitting projected silhouette edges according to vertical coordinates values of end points and intersect points of these edges; (5)Constructing trapeziums to determine the visibility of light source. In the following paragraph we explain the details for each step.

First of all, all global potential silhouette edges should be found out. The method is very similar with that described in [1]. It is simple and clear as shown in Fig.1.

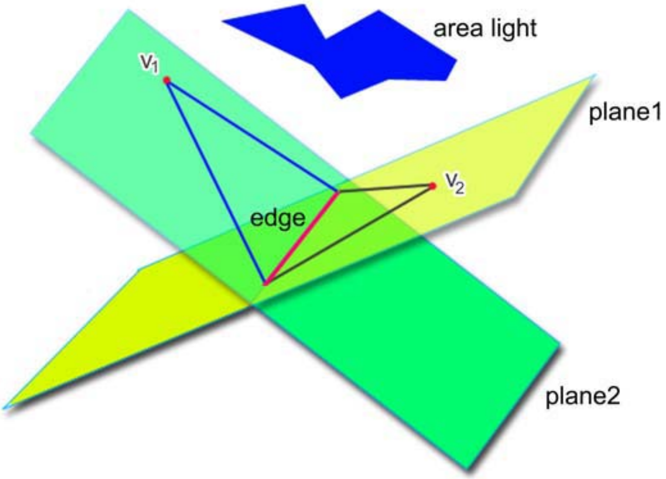


Fig. 1. Demonstrating how to choose global silhouette edges from designated area light source (the blue polygon). The red line segment is an edge. The two triangles connected to the edge lie on *plane1* and *plane2* respectively. When *area light* and v_1 are on the same side of *plane1* and *area light* and vertex v_2 are on the same side of *plane2*, the edge isn't silhouette edge; When *area light* and v_1 are on the different side of *plane1* and *area light* and vertex v_2 are on the different side of *plane2*, the edge isn't silhouette edge; Otherwise the edge is silhouette edge. If there is only a single triangle connected to an edge, it must always be considered as a global potential silhouette edge.

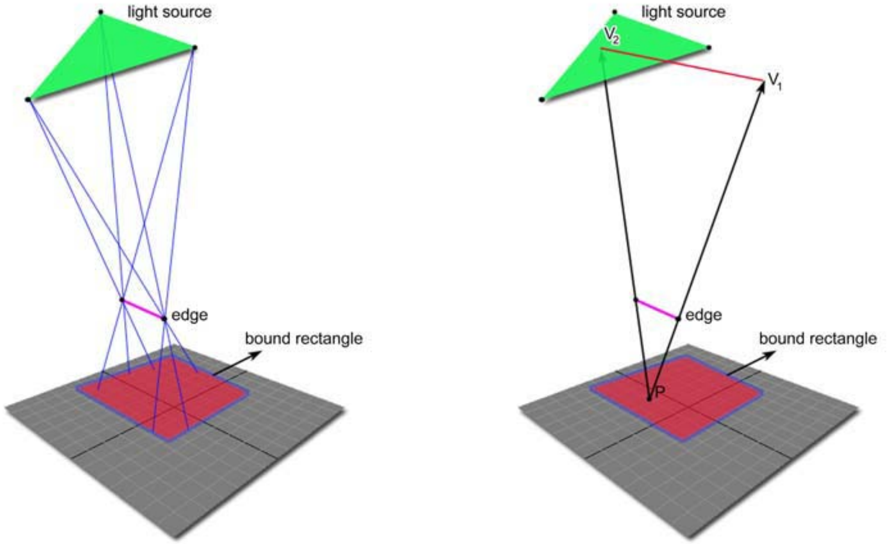


Fig. 2. The first picture illustrate the process of constructing bound rectangle of projected vertices of light source. The second picture determines whether light source is visible from a scene point P.

For a given shadow point *P*, only a small proportion of the global potential silhouette edges are silhouette edges. In order to a quickly find out the relevant silhouette edges for *P,* we store all global potential silhouette edges into a grid data structure according to bound rectangle of projected light source on scene plane. Each shadow point maintains a list used to keep all relevant silhouette edges. For each global potential silhouette edge we construct a bound rectangle using projected vertices of light source as shown in the first picture of Fig.2. We add the edge to the lists of shadow points which lie inside the rectangle. After all global potential silhouette edges have been enumerated, for each shadow point *P* its list contains all local potential silhouette edges relevant to it.

We now turn to the process of the soft shadow query for each shadow point. The query is issued for a given shadow point *P* and determines how much proportion of the light source is visible from *P*. For each *P*, The first step of the query is to enumerate its list to determine which silhouette edge intersect with the light source as seen from *P* (second picture of Fig.2). Then, we test that these edges are silhouette edges from *P*. We can use the method described in Fig.1 to determine if an edge is a silhouette edge from *P* [2, 11].

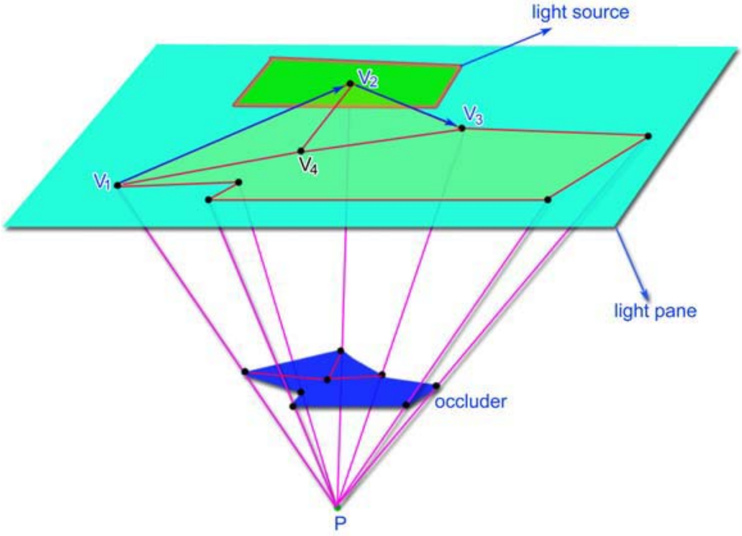


Fig. 3. Projecting silhouette edges to the light source plane from *P*. Only edges which lie inside light source or intersect with light source will be used to calculate soft shadow volumes.

After collecting the list of exact silhouette edges from *P*, we project all these silhouette edges to the light source plane from *P* (Fig.3). The projected silhouette edges are oriented according to the relative position of the triangle containing the edge. We let the orientation of projected silhouette edge be *ascent* if the triangle lies on right of the edge, otherwise it is *descent*. As illustrated in Fig.3, the orientation of edge V_1V_2 is *ascent* and V_2V_3 is *descent*. The orientation property of silhouette edge is very important when calculating visibility of light source. Some projected silhouette edges should be modified if they intersect with the left side of light source (Fig.4). Fig.4 shows the six cases that need to modify silhouette edges. The first four cases have been described in [1], we extend the rules used in [1] by Fig.4e and Fig.4f.

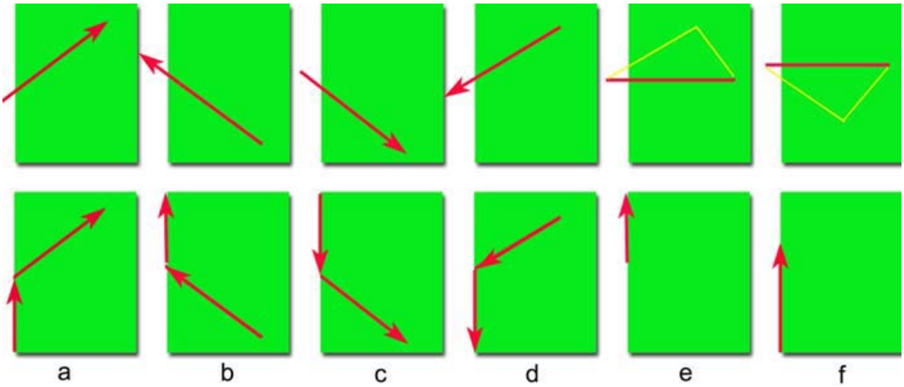


Fig. 4. The green rectangles represent the area light source. The red lines are silhouette edges. If a silhouette edge intersects with the left side of light source, it should be modified. The first row of this picture shows the original edge and the second row show the modified edge. a, b, c and d break the original edge into two edges. e and f show how to deal with horizontal edges. In e and f the yellow lines and the horizontal edge construct a projected triangle of occluder. *e* shows the case when the triangle is above the edge and f shows the case when the triangle is under the edge. In e the converted edge is above the original one and in f the converted edge is under the original one. The properties of changed edges keep the same as original. In *a*, *b*, can *d* a vertical silhouette edges is added. In *e* and *f* the original horizontal edges are discarded.

Now we split all silhouette edges according to intersect points and end points of silhouette edges in vertical coordinate. Fig.5 shows the process of splitting. In the second picture of Fig.5, edges will be broken at the red points. The broken edges will keep the same orientation property as the original ones. After splitting, the light source is partitioned to many small trapeziums (Fig.6). Each trapezium has unique visibility property.

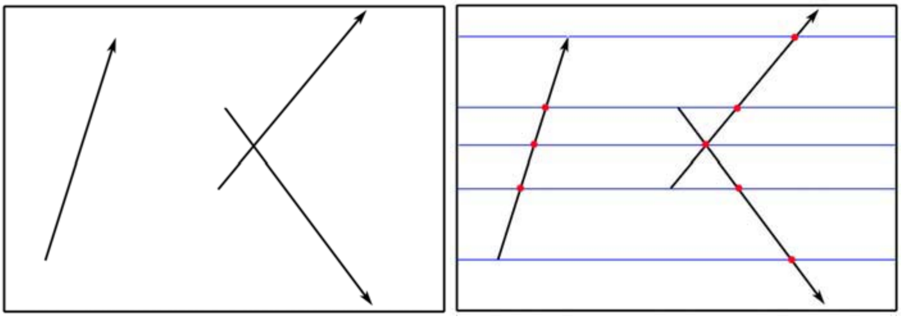


Fig. 5. Splitting silhouette edges according to end points and intersect points of all edges in vertical coordinate. The first picture shows the original edges and the second picture shows the red break points.

From the definition of the orientation property of silhouette edge we can find that an *ascent* silhouette edge indicates the beginning of an invisible trapezium occluded by occluders on light source and a *descent* silhouette edge means the ending of the region. Let the number of surfaces which occlude the light source be *depth* number [1]. We can mark all trapeziums by *depth* numbers. Each trapezium has a *depth* number to indicate its visibility status. When calculating *depth* numbers, the initial value for each line is 0 (see Fig.6). Then we calculate the *depth* numbers from left to right. For a trapezium its *depth* number is equals to the previous *depth* number plus an integer *T* when the left side of the trapezium is an *ascent* silhouette edge; otherwise we subtract *T* from the previous *depth* number. *T* is 1 if there is only a single triangle connected to the silhouette edge, otherwise *T* is 2. In Fig.6, all *depth* numbers is calculated in case that *T* is 1. Special care must be taken to account for the missing silhouette edges outside the light source. When projected silhouette edges lie on the left side of the light source they are omitted. The methods shown in Fig.4 can deal with all these incorrect cases. From the methods shown in Fig.4 we can see that the *depth* number of trapezium is a relative value. The trapeziums with the lowest *depth* number may be invisible from shadow points because those projected silhouette edges on the left side of the light source are omitted. We must test these trapeziums to determine if they are visible. In order to accomplish this, we cast a single reference ray from shadow point *P* to the barycenter of one of these trapeziums and count the number of surfaces it intersects with the occluders, yielding the actual *depth* number of these areas. If the actual depth number is greater than 0 the whole light source is invisible. Otherwise, we can get the visibility rate of the light source as:

$$R=1-\sum s_t/S_l$$

where R is *visibility rate* for a shadow point, S_t is the area of trapezium with lowest depth number and S_l is the area of the whole light source.

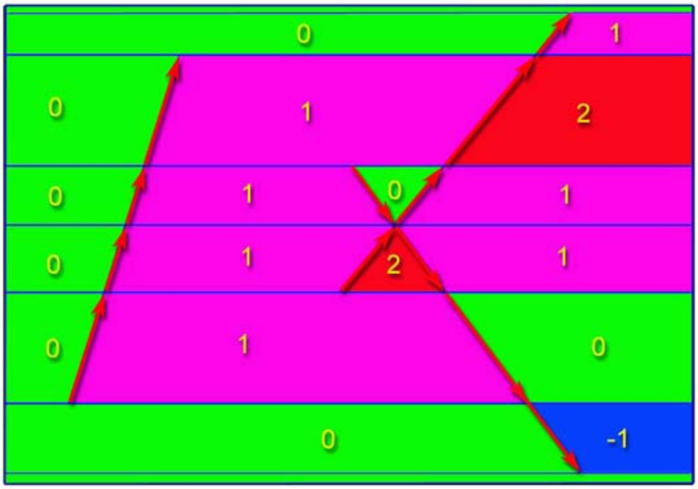


Fig. 6. Trapeziums are constructed according to the property of silhouette edge. *Depth* number of each trapezium is calculated and trapeziums with different depth number are drawn using different color. In this figure *T* always is 1.

Using the method describe previously we can numerate all shadow points to get their visibility rate of light source and then draw the soft shadow quickly. The render resolution of the examples of this paper is 200×200. We interpolate *visibility rate* between two neighboring shadow points linearly.

3 Results and Conclusions

In this section, we consider several examples of soft shadows produced using our prototype. Our prototype is implemented by Visual C++. All results were generated on a PC with P4 2.66 GHZ CPU, 1G RAM and Windows OS. In Fig.7, 8, 9 and 10, the major strength of our algorithm is shown, namely that soft shadows can be cast on a complex planar shadow receiver. As can be seen, the rendered images exhibit typical characteristics of soft shadows: the shadows are softer the farther away the occluder is from the receiver, and they are hard where the occluder is near the receiver.

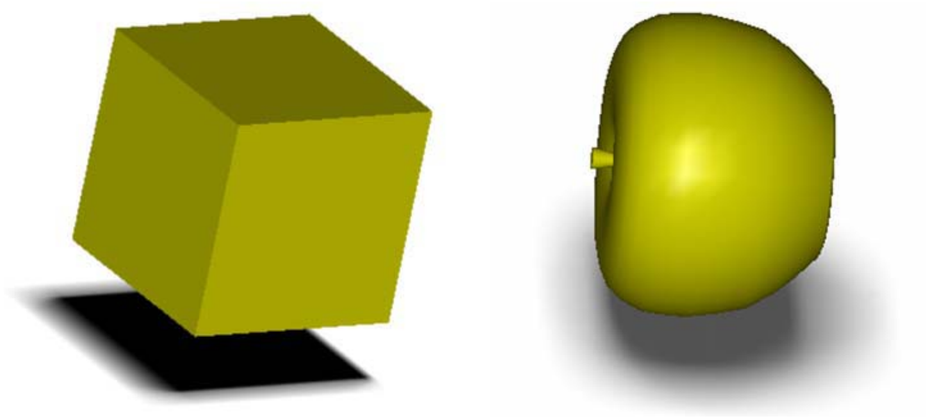


Fig. 7. Soft shadows of a cubic and an apple are cast to a white floor

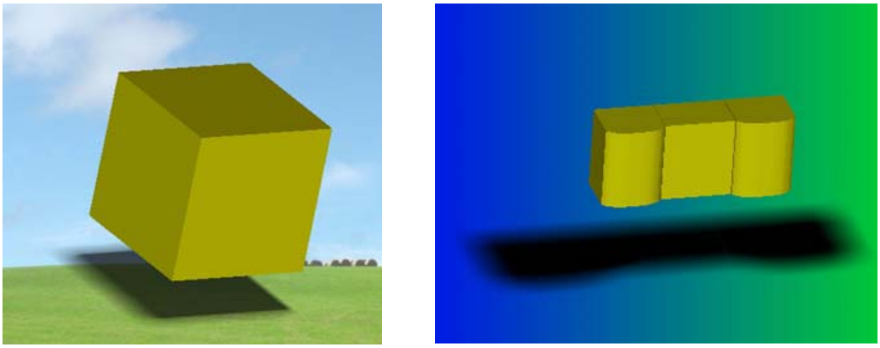


Fig. 8. Soft shadows of a cubic and a box blend with complex background

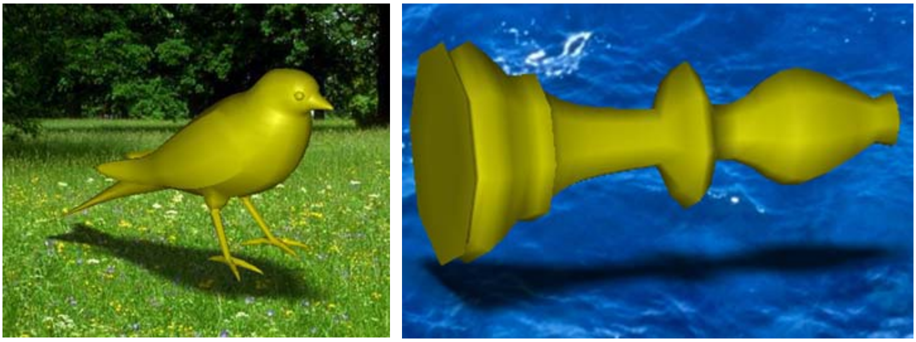


Fig. 9. More complex meshes (bishop and bird) create soft shadows on different nature scenes

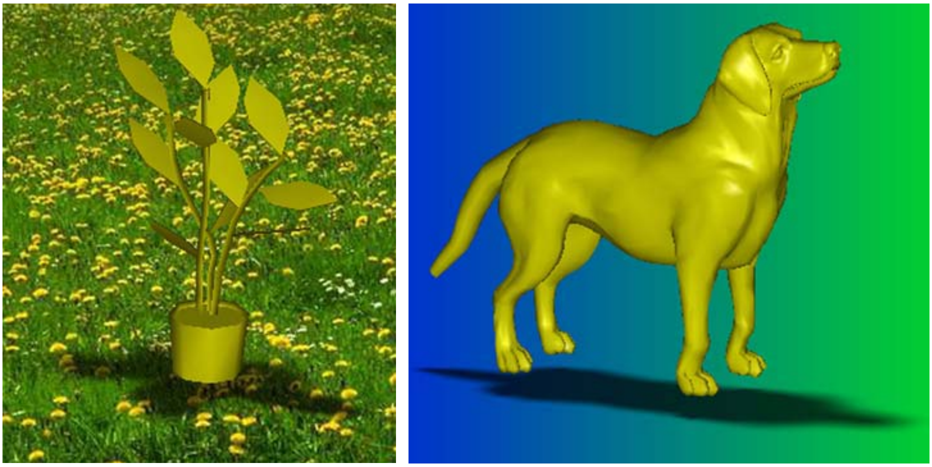


Fig. 10. A large dog mesh is used to test the strength of method. A flower is placed to a lawn naturally.

Table 1. Performance results from examples (Fig.7-10) with a scene resolution of 200×200. All computations performed by our algorithm are included in the timings. The times (in seconds) are gotten on a PC with 1G RAM and 2.66GHZ CPU.

Meshes	*# of*	*# of*	*Distance to*	*Size of*	*Time*
	vertices	*triangles*	*light(cm)*	*light(cm)*	*(s)*
bishop	250	496	50	5.0×5.0	1.011
apple	891	1704	50	3.0×3.0	2.012
flower	242	401	100	1.0×1.0	0.934
bird	1155	2246	30	5.0×5.0	3.234
cube	8	12	20	8.0×8.0	0.005
dog	6650	13176	50	3.0×3.0	18.126
box	120	228	30	10.0×10.0	0.082

Our new algorithm offers a significant performance improvement in a variety of test scenes. The proposed method in this paper is suitable for computing shadows from very large light sources because it does not require additional light samples for limiting the noise to an acceptable level as the method used in [1].We compare our method with the method presented in [1]. Our method is faster when the quality of results is same. While our research takes a small step down to a new path, several limits remain as future work. The most conspicuous limitation of our technique is that for large scene resolution our method is also expensive like other soft shadow algorithms and need use intelligent ways to overcome it. As future work, we would like to investigate better sampling strategies of scene.

Acknowledgements

The authors would like to thank the support from the China NSF under grant #60473106, China Ministry of Education under grant#20030335064, China Ministry of Science and Technology under grant #2003AA4Z3120.

References

1. Laine, S., Aila, T., Assarsson, U., Lehtinen, J., and Akenine-Möller, T. 2005. Soft shadow volumes for ray tracing. *ACM Trans. Graph.* 24, 3 (Jul. 2005), 1156-1165.
2. Brabec S., Seidel H.-P.: Shadow volumes on programmable graphics hardware. In Proceedings of Eurographics (2003), vol. 22, pp. 433--440. 6
3. Assarsson, U. and Akenine-Möller, T. 2003. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Trans. Graph.* 22, 3 (Jul. 2003), 511-520.
4. Akenine-Möller, T. and Assarsson, U. 2002. Approximate soft shadows on arbitrary surfaces using penumbra wedges. In *Proceedings of the 13th Eurographics Workshop on Rendering* (Pisa, Italy, June 26 - 28, 2002). S. Gibson and P. Debevec, Eds. ACM International Conference Proceeding Series, vol. 28. Eurographics Association, Aire-la-Ville, Switzerland, 297-306.
5. AndrewWoo, Pierre Poulin, and Alain Fournier. A survey of shadow algorithms. *IEEE Computer Graphics and Applications*, 10(6):13–32, Nov. 1990.
6. Hasenfratz, J.-M., Lapierre, M., Holzschuch, N., and Sillion, F. 2003. A Survey of Real-Time Soft Shadows Algorithms. Computer Graphics Forum 22, 4, 753–774.
7. Akenine-M¨OLLER, T., and Assarsson, U. 2002. Approximate Soft Shadows on Arbitrary Surfaces using Penumbra Wedges. In 13th Eurographics Workshop on Rendering, Eurographics, 297–305.
8. Turner Whitted. An Improved Illumination Model for Shaded Display. Communications of the ACM, 23:343–349, 1980.
9. Amanatides, J. 1984. Ray Tracing with Cones. In Computer Graphics (Proceedings of ACM SIGGRAPH 84), ACM Press, 129–135.
10. Stefan Brabec and Hans-Peter Seidel. Single sample soft shadows using depth maps. In Graphics Interface, 2002.
11. MCGUIRE, M. 2004. Observations on Silhouette Sizes. Journal of Graphics Tools 9, 1, 1–12.

Photo-Consistent Motion Blur Modeling for Realistic Image Synthesis

Huei-Yung Lin and Chia-Hong Chang

Department of Electrical Engineering,
National Chung Cheng University,
168 University Rd., Min-Hsiung
Chia-Yi 621, Taiwan, R.O.C.
lin@ee.ccu.edu.tw, g93415060@ccu.edu.tw

Abstract. Motion blur is an important visual cue for the illusion of object motion. It has many applications in computer animation, virtual reality and augmented reality. In this work, we present a nonlinear imaging model for synthetic motion blur generation. It is shown that the intensity response of the image sensor is determined by the optical parameters of the camera and can be derived by a simple photometric calibration process. Based on the nonlinear behavior of the image intensity response, photo-realistic motion blur can be obtained and combined with real scenes with least visual inconsistency. Experiments have shown that the proposed method generates more photo-consistent results than the conventional motion blur model.

1 Introduction

In the past few years, we have witnessed the convergence of computer vision and computer graphics [1]. Although traditionally regarded as inverse problems of each other, image-based rendering and modeling shares the common ground of these two research fields. One of its major topics is how to synthesize computer images for graphics representations based on the knowledge of a given vision model. This problem commonly arises in the application domains of computer animation, virtual reality and augmented reality [2]. For computer animation and virtual reality, synthetic images are generated from existing graphical models for rendering purposes. Augmented reality, on the other hand, requires the image composition of virtual objects and real scenes in a natural way. The ultimate goal of these applications is usually to make the synthetic images look as realistic as those actually filmed by the cameras.

Most of the previous research for rendering synthetic objects into real scenes deal with static image composition, even used for generating synthetic video sequences. When modeling a scene containing a fast moving object during the finite camera exposure time, it is not possible to insert the object directly into the scene by simple image overlay. In addition to the geometric and photometric consistency imposed on the object for the given viewpoint, motion blur or temporal aliasing due to the relative motion between the camera and the scene usually have to be taken into account. It is a very important visual cue to human

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1273–1282, 2006.

perception for the illusion of object motion, and commonly used in photography to illustrate the dynamic features in the scene. For computer generated or stop motion animations with limited temporal sampling rate, unpleasant effects such as jerky or strobing appearance might present in the image sequence if motion blur is not modeled appropriately.

Early research on the simulation of motion blur suggested a method by convolving the original image with the linear optical system-transfer function derived from the motion path [3,4]. The uniform point spread function (PSF) was demonstrated in their work, but high-degree resampling filters were later adopted to further improve the results of temporal anti-aliasing [5]. More recently, Sung et al. introduced the visibility and shading functions in the spatial-temporal domain for motion blur image generation [6]. Brostow and Essa proposed a frame-to-frame motion tracking approach to simulate motion blur for stop motion animation [7]. Except for the generation of realistic motion blur, there are also some researchers focusing on real-time rendering using hardware acceleration for interactive graphics applications [8,9]. Although the results are smooth and visually consistent, they are only approximations due to the oversimplified image formation model.

It is commonly believed that the image acquisition process can be approximated by a linear system, and motion blur can thus be obtained from the convolution with a given PSF. However, the nonlinear behavior of image sensors becomes prominent when the light source changes rapidly during the exposure time [10]. In this case, the conventional method using a simple box filter cannot create the photo-realistic or photo-consistent motion blur phenomenon. This might not be a problem in purely computer-generated animation, but inconsistency will certainly be noticeable in the image when combining virtual objects with real scenes. Thus, in this work we have proposed a nonlinear imaging model for synthetic motion blur generation. Image formation is modified and incorporated with nonlinear intensity response function. More photo-consistent simulation results are then obtained by using the calibrated parameters of given camera settings.

2 Image Formation Model

The process of image formation can be determined by the optical parameters of the lens, geometric parameters of the camera projection model, and photometric parameters associated with the environment and the CCD image sensor. To synthesize an image from the same viewpoint of the real scene image, however, only the photometric aspect of image formation has to be considered. From basic radiometry, the relationship between scene radiance L and image irradiance E is given by

$$E = L\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4 \alpha \qquad (1)$$

where d, f and α are the aperture diameter, focal length and the angle between the optical axis and the line of sight, respectively [11]. Since the image intensity is commonly used to represent the image irradiance, it is in turn assumed proportional to the scene radiance for a given set of camera parameters. Thus, most existing algorithms adopt a simple pinhole camera model for synthetic image generation of real scenes. Linear motion blur is generated by convolving the original image with a box filter or a uniform PSF [3]. Although the image synthesis or composition are relatively easy to implement based on the above image formation, the results are usually not satisfactory when compared to the real images captured by a camera. Consequently, photo-realistic scene modeling cannot be accomplished by this simplified imaging model.

One major issue which is not explicitly considered in the previous approach is the nonlinear behavior of the image sensors. It is commonly assumed that the image intensity increases linearly with the camera exposure time for any given scene point. However, nonlinear sensors are generally designed to have the output voltage proportional to the log of the light energy for high dynamic range imaging [12,13]. Furthermore, the intensity response function of the image sensors is also affected by the F-number of the camera from our observation.

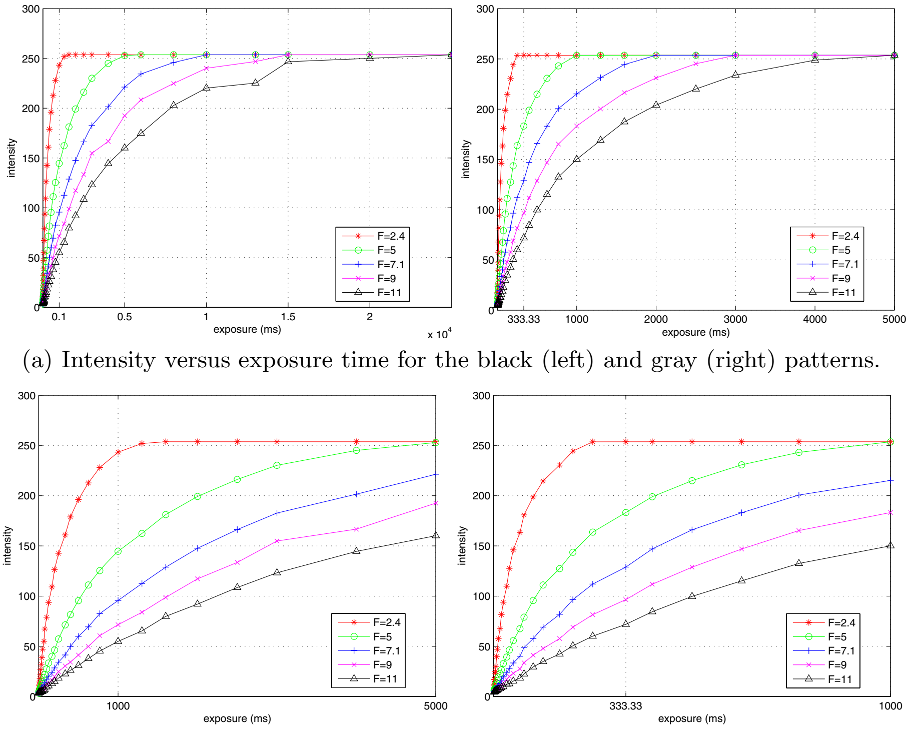
To illustrate this phenomenon, an image printout with white, gray and black stripes is used as a test pattern. Image intensity values under different camera exposures are calibrated for various F-number settings. The plots of intensity value versus exposure time for both the black and gray image stripes[1] are shown in Figure 1. The figures demonstrate that, prior to saturation, the intensity values increase nonlinearly with the exposure times. Although the nonlinear behaviors are not severe for large F-numbers (i.e., small aperture diameters), they are conspicuous for smaller F-numbers. Another important observation is that, even with different scene radiance, the intensity response curves for the black and gray patterns are very similar if the time axis is scaled by a constant. Figure 2 shows the intensity response curves for several F-numbers normalized with respect to the gray and black image patterns. The results suggest that the intensity values of a scene point under different exposure time are governed by the F-number.

To establish a more realistic image formation model from the above observations, a monotonically increasing function with nonlinear behavior determined by additional parameters should be adopted. Since the intensity response curves shown in Figure 1 cannot be easily fitted by gamma or log functions with various F-numbers, we model the intensity accumulation versus exposure using an operation similar to the capacitor charging process. The intensity value of an image pixel $I(t)$ is modeled as an inverse exponential function of the integration time t given by

$$I(t) = I_{\max}(1 - e^{-k\frac{d^2}{\rho}t}) \qquad \text{for} \quad 0 \leq t \leq T \tag{2}$$

where T, $I_{\max}$, d, k and ρ are the exposure time, maximum intensity, aperture diameter, a camera constant, and a parameter related to the object surface's reflectance property, respectively. If all the parameters in Eq. (2) are known,

[1] The intensity response of the white image pattern is not shown because it is saturated in a small exposure range.



(a) Intensity versus exposure time for the black (left) and gray (right) patterns.

(b) Small exposure range clearly shows the nonlinear behavior of the intensity.

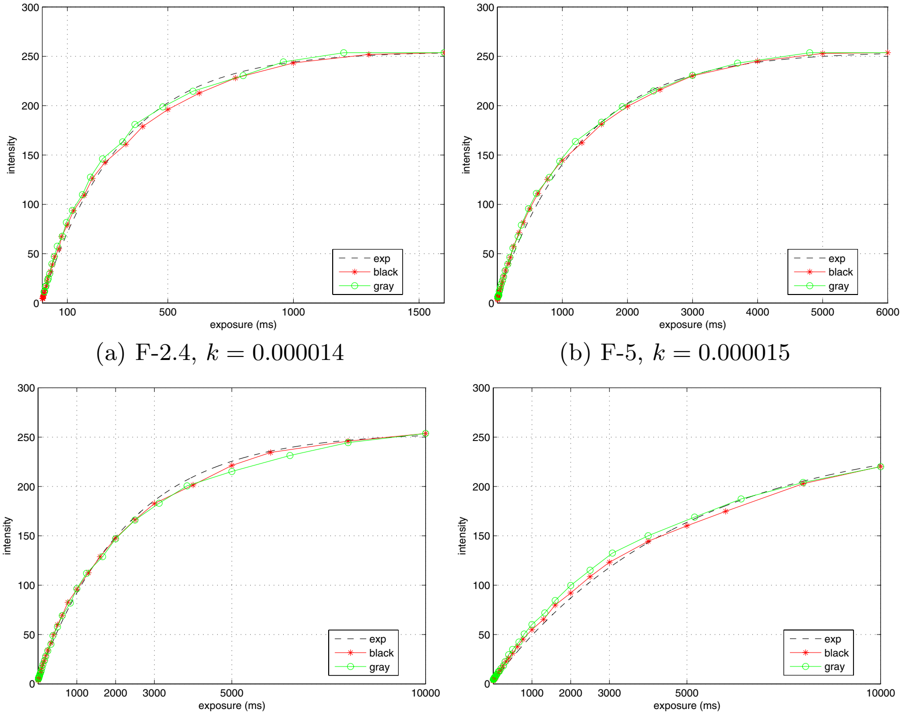
Fig. 1. Nonlinear behavior of the intensity response curves with different F-numbers

then it is possible to determine the intensity value of the image pixel for any exposure time less than T.

For a general 8-bit greyscale image, the maximum intensity $I_{\max}$ is 255 and $I(t)$ is always less than $I_{\max}$. The aperture diameter d is defined as the F-number divided by the focal length, and can be obtained from the camera settings. The parameters k and ρ are constants for any fixed scene point in the image. Thus, Eq. (2) can be rewritten as

$$I(t) = I_{\max}(1 - e^{-k't}) \tag{3}$$

for a given set of camera parameters. The only parameter k' can then be determined by an appropriate calibration procedure with different camera settings. To verify Eq. (3), we first observe that $I(0) = 0$ as expected for any camera settings. The intensity value saturates as $t \rightarrow \infty$, and the larger the parameter k' is, the faster the intensity saturation occurs. This is consistent with the physical model: k' contains the reflectance of the scene point and thus represents the irradiance of the image point. Figures 1 and 2 illustrate that, the nonlinear responses are not noticeable for small apertures, but they are evident for large aperture sizes. For either case, the response function can be modeled by Eq. (3)



(a) F-2.4, $k = 0.000014$

(b) F-5, $k = 0.000015$

(c) F-7.1, $k = 0.0000166$

(d) F-11, $k = 0.000019$

Fig. 2. Normalized intensity response curves for different F-numbers

with some constant k'. Thus, the most important aspect of the equation is to characterize the image intensity accumulation versus integration time based on the fixed camera parameters.

For a given intensity value, it is not possible to determine the exposure time since the image irradiance also depends on the object's reflectance property. However, it is possible to calculate the image intensity of a scene point under any exposure if an intensity-exposure pair is given and the normalized response curve is known for specific camera parameter settings. This is one of the requirements for generating space-variant motion blur as described in the following section.

To obtain the normalized intensity response function up to an unknown scale factor in the time domain, the images of the calibration patterns are captured with various exposure followed by least-squared fitting to find the parameter k' for different F-numbers. As shown in Figure 2, the resulting fitting curves (black dashed lines) for any given F-number provide good approximation to the actual intensity measurements for both the black and gray patterns. This curve fitting and parameter estimation process can be referred to as photometric calibration for the intensity response function. It should be noted that only the shape of the intensity response curve is significant, the resulting function is normalized in the time axis by an arbitrary scale factor. Given the intensity value of an image pixel with known camera exposure, the corresponding scene point under different amount of exposure can be calculated by Eq. (3).

3 Synthetic Motion Blur Image Generation

Motion blur arises when the relative motion between the scene and the camera is fast during the exposure time of the imaging process. The most commonly used model for motion blur is given by

$$g(x, y) = \int_0^T f(x - x_0(t), y - y_0(t))dt \tag{4}$$

where $g(x, y)$ and $f(x, y)$ are the blurred and ideal images, respectively. T is the duration of the exposure. $x_0(t)$ and $y_0(t)$ are the time varying components of motion in the x and y directions, respectively [3]. If only the uniform linear motion in the x-direction is considered, the motion blurred image can be generated by taking the average of line integral along the motion direction. That is,

$$g(x, y) = \frac{1}{R} \int_0^R f(x - \rho, y)d\rho \tag{5}$$

where R is the extent of the motion blur. Eq. (5) essentially describes the blurred image as the convolution of the original (ideal) image with a uniform PSF

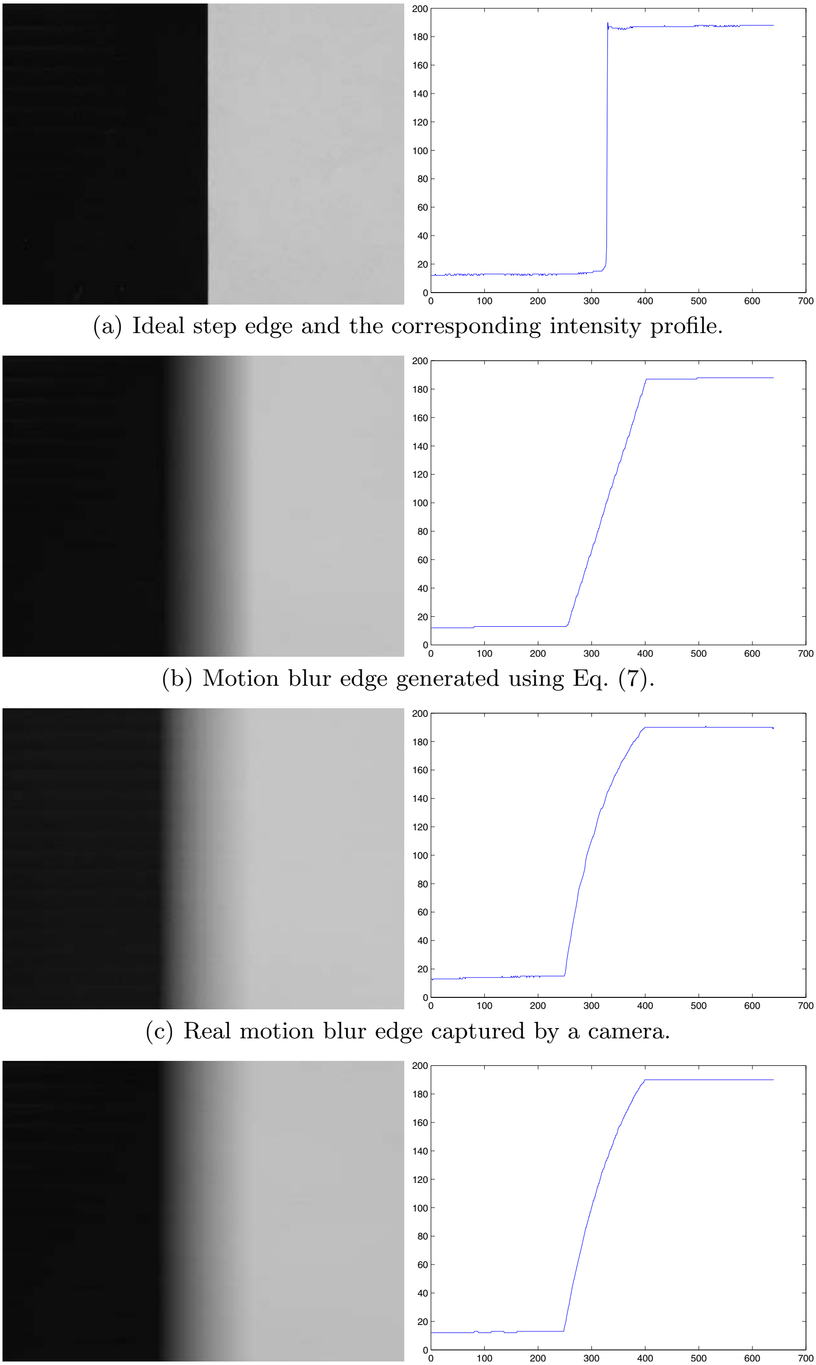
$$h(x, y) = \begin{cases} 1/R, & |x| \le R/2 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

This model is de facto the most widely adopted method for generating motion blur images. Its discrete counterpart used for computation is given by

$$g[m, n] = \frac{1}{K} \sum_{i=0}^{K-1} f[m - i, n] \tag{7}$$

where K is the number of blurred pixels.

As an example of using the above image degradation model, motion blur of an ideal step edge can be obtained by performing the spatial domain convolution with the PSF given by Eq. (6). The synthetic result is therefore a ramp edge with the width of the motion blur extent R. If this motion blur model is applied on a real edge image, however, the result is generally different from the recorded motion blur image. Figures 3(a), 3(b) and 3(c) illustrate the images and intensity profiles of an ideal step edge, motion blur edge created using Eq. (7) and real motion blur edge captured by a camera, respectively. As shown in Figure 3(c), the intensity profile indicates that there exists non-uniform weighting on the pixel intensities of real motion blur. Since the curve is not symmetric with respect to its midpoint, this nonlinear response is clearly not due to the optical defocus of the camera and cannot be described by a Gaussian process.



(a) Ideal step edge and the corresponding intensity profile.

(b) Motion blur edge generated using Eq. (7).

(c) Real motion blur edge captured by a camera.

(d) Motion blur edge synthesized by the proposed method.

Fig. 3. Motion blur synthesis of an ideal edge image

In this work, motion blur is modeled using nonlinear intensity response of the image sensor as discussed in the previous section. For an image position under uniform motion blur, its intensity value is given by the integration of image irradiance associated with different scene points during the exposure time. Every scene point in the motion path thus contributes the intensity for smaller yet equal exposure time. Although these partial intensity values can be derived from the static image with full exposure by linear interpolation in the time domain, nonlinear behavior of the intensity response should also be taken into account. Suppose the monotonic response function is $I(t)$, then the motion blur image $g(x, y)$ is given by

$$g(x,y) = I\left(\frac{1}{R}\int_0^R I^{-1}(f(x-\rho,y))d\rho\right) \tag{8}$$

where R is the motion blur extent and $I^{-1}(\cdot)$ is the inverse function of $I(t)$. The discrete counterpart of Eq. (8) is given by

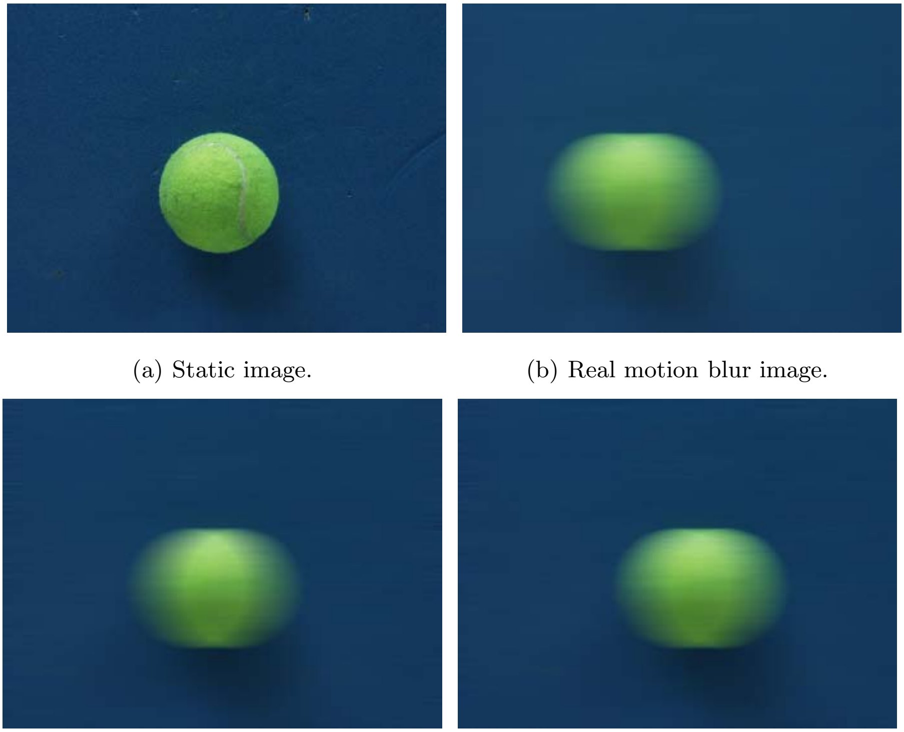
$$g[m,n] = I\left(\frac{1}{K}\sum_{i=0}^{K-1} I^{-1}(f[m-i,n])\right) \tag{9}$$

where K is the number of blurred pixels. If we consider the special case that the intensity response function $I(t)$ is linear, then Eqs. (8) and (9) are simplified to Eqs. (5) and (7), respectively.

Figure 3(d) shows the synthetic motion blur edge of Figure 3(a) and the corresponding intensity profile of the image scanlines generated using Eq. (9). The intensity response function $I(t)$ is given by Eq. (3) with F-5 and $k' = 0.000015$. By comparing the generated images and intensity profiles with those given by the real motion blur, the proposed model clearly gives more photo-consistent results than the one synthesized using uniform PSF. The fact that brighter scene points contribute more intensity to the image pixels, as shown in Figure 3(c), is successfully modeled by the nonlinear response curve.

4 Results

Figure 4 shows the experimental results of a real scene. The camera is placed at about 1 meter in front of the object (a tennis ball). The static image shown in Figures 4(a) is taken at F-5 with exposure time of 1/8 second. Figure 4(b) shows the motion blur image taken under 300 mm/sec. lateral motion of the camera using the same set of camera parameters. The blur extent in the image is 180 pixels, which is used for synthetic motion blur image generation. Figure 4(c) illustrates the motion blur synthesized using the widely adopted uniform PSF for image convolution. The result generated using the proposed nonlinear intensity response function is shown in Figure 4(d). For the color images, red, green and blue channels are processed separately. Motion blur images are first created for each channel using the same intensity response curve, and then combined to form the final result.



(a) Static image.

(b) Real motion blur image.

(c) Motion blur generated using Eq. (7).

(d) Motion blur by the proposed method.

Fig. 4. Experimental results of a real scene

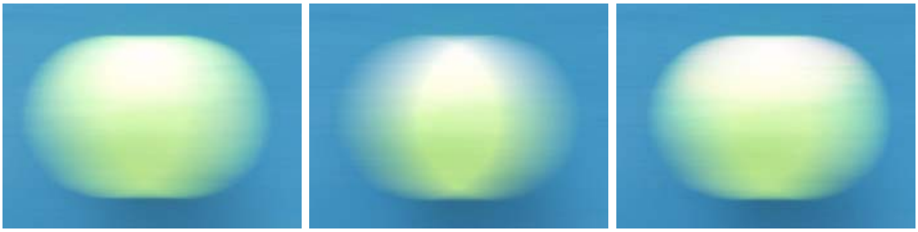


Fig. 5. Motion blur generated with small F-number (large aperture size)

With careful examination of Figure 4, it is not difficult to find that the image shown in Figure 4(d) is slightly better than Figure 4(c). The image scanline intensity profiles of Figure 4(d) are very close to those exhibited in the real motion blur image. Figure 5 (left) shows another example taken at F-2.4 with exposure time of 1/8 second. The middle and right figures are the results using Eq. (7) and the proposed method, respectively. It is clear that the nonlinear behavior becomes prominent and has to be considered for more realistic motion blur synthesis.

5 Conclusion

Image synthesis or composition with motion blur phenomenon have many applications in computer graphics and visualization. Most existing works generate motion blur by convolving the image with a uniform PSF. The results are usually not photo-consistent due to the nonlinear behavior of the image sensors. In this work, we have presented a nonlinear imaging model for synthetic motion blur generation. More photo-realistic motion blur can be obtained and combined with real scenes with least visual inconsistency. Thus, our approach can be used to illustrate dynamic motion for still images, or render fast object motion with limited frame rate for computer animation.

References

1. Lengyel, J.: The convergence of graphics and vision. Computer **31**(7) (1998) 46–53
2. Kutulakos, K.N., Vallino, J.R.: Calibration-free augmented reality. IEEE Transactions on Visualization and Computer Graphics **4**(1) (1998) 1–20
3. Potmesil, M., Chakravarty, I.: Modeling motion blur in computer-generated images. In: Proceedings of the 10th annual conference on Computer graphics and interactive techniques, ACM Press (1983) 389–399
4. Max, N.L., Lerner, D.M.: A two-and-a-half-d motion-blur algorithm. In: SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM Press (1985) 85–93
5. Dachille, F., Kaufman, A.: High-degree temporal antialiasing. In: CA '00: Proceedings of the Computer Animation. (2000) 49–54
6. Sung, K., Pearce, A., Wang, C.: Spatial-temporal antialiasing. IEEE Transactions on Visualization and Computer Graphics **08**(2) (2002) 144–153
7. Brostow, G., Essa, I.: Image-based motion blur for stop motion animation. In: SIGGRAPH 01 Conference Proceedings, ACM SIGGRAPH (2001) 561–566
8. Wloka, M.M., Zeleznik, R.C.: Interactive real-time motion blur. The Visual Computer **12**(6) (1996) 283–295
9. Meinds, K., Stout, J., van Overveld, K.: Real-time temporal anti-aliasing for 3d graphics. In Ertl, T., ed.: VMV, Aka GmbH (2003) 337–344
10. Rush, A.: Nonlinear sensors impact digital imaging. Electronics Engineer (1998)
11. Forsyth, D., Ponce, J.: Computer Vision: A Modern Approach. Prentice-Hall (2003)
12. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ACM Press (1997) 369–378
13. Schanz, M., Nitta, C., Bussman, A., Hosticka, B.J., Wertheimer, R.K.: A high-dynamic-range cmos image sensor for automotive applications. IEEE Journal of Solid-State Circuits **35**(7) (2000) 932–938

Integration of GPS, GIS and Photogrammetry for Texture Mapping in Photo-Realistic City Modeling

Jiann-Yeou Rau[1], Tee-Ann Teo[2], Liang-Chien Chen[3], Fuan Tsai[4],
Kuo-Hsin Hsiao[5], and Wei-Chen Hsu[5]

[1] Specialist, [2] Ph. D Student, [3] Professor, [4] Assistant Professor,
Center for Space and Remote Sensing Research,
National Central University (NCU), Jhong-Li, Taiwan
{jyrau, ann, lcchen, ftsai}@csrsr.ncu.edu.tw
[5] Researcher, Energy and Environment Lab.,
Industrial Technology Research Institute (ITRI), Chu-Tung, Taiwan
{HKS, ianhsu}@itri.org.tw

Abstract. The generation of the textures of building facades from terrestrial photos is time-consuming work. A more cost-effective way to create a photo-realistic city model containing thousands of buildings and facade textures is necessary. This paper proposes an approach which integrates GPS, GIS and photogrammetry for multi-face texture mapping of a three-dimensional building model. A GPS integrated, high-resolution, non-metric and wide field-of-view digital camera is used. By means of integrating the GIS graphic interface and GPS information about camera location, a large quantity of pictures can be managed efficiently. A graphics user interface for the interactive solving of the exterior orientation parameters is designed. In the meantime, during texturing mapping lens-distortion can be corrected and self-occluded facades can be automatically compensated for. The experimental results indicate that the proposed approach is efficient due to a complex building could be treated in one process and multi-face texturing is designed thus the number of pictures can be reduced. Additionally, a large quantity of pictures can be managed efficiently by GPS-Photo-GIS integration.

Keywords: Photo-Realistic Modeling, Texture Mapping, City Model.

1 Introduction

The presentation of a photo-realistic 3D city model is of major interest in the field of geoinformatics. On the city scale, a large number of building models and terrestrial photos need to be treated. In the meantime, the occlusion problems may be introduced by the buildings itself (i.e. self-occlusion) or due to objects in front of the buildings, such as cars, trees, street signs, etc. Thus, high performance production throughput, accurate geometric quality and occlusion-free texture generation are three of the major problems to solve.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1283 – 1292, 2006.

1.1 Related Work

A number of algorithms for the purpose of the extraction of 3D geometric building models have already been reported. These algorithms can be categorized as either an automatic strategy [1] or a semi-automatic approach [2]. The representation of 3D building models is a generalization procedure. This means that from a city scale point of view, it is impractical to describe complex buildings using detailed geometrical descriptions. Since a detailed sketch of the buildings includes the windows, ornaments, doors and so on, will introduce a huge volume of data, which cause problem in real applications. Such detailed structures can be omitted when far field visualization is considered, however in near field applications these detailed structures are important for realization and recognition.

Facade texture provides complementary information for a generalized building model but detailed geometric structures are lost. The texture provides a photo-realistic effect and conveys the visually-detailed geometrical structure of a building. Such details are getting more important for many simulation applications, such as urban planning, virtual tourism, the digital documenting of historic landmarks, and so on. The creation of rooftop texture can be acquired from aerial photos by a true-orthophoto generation process [3]. However, this is not suitable for the generation of facade texture due to the large amount of perspective distortion.

Brenner & Haala [4] have proposed an interactive scheme which utilizes projective transformation for the fast production of building facade textures. Haala [5] has proposed using a high-resolution panoramic CCD line scanner to improve the efficiency of texture collection. Since one picture can cover a large quantity of buildings, the overall effort needed can be reduced. However, this method requires an elevated camera station to minimize occlusions, and an expensive rotating CCD scanner.

Klinec and Fritsch [6] determined the camera's exterior orientation parameters by searching for correspondences between building models and image features automatically. The results are used in a photogrammetric space resection process. The occlusion-free facade texture can be fused from multi-views of images. Varshosaz [7] proposed an automatic approach utilizing a specially designed photo system with a servo-driven theodolite. The system can take images automatically in a step-wise manner at preset locations covering the complete field of view. The theodolite angles are used to index image, thus only one space resection has to be done for each station. The texturing of the final facade is a combination of images from different views.

1.2 Concept of the Paper

In the paper, we describe an efficient and good quality method using a high-resolution, wide field-of-view (FOV), GPS integrated digital camera for texture collection. The wide FOV camera used will introduce significant lens distortion. So, the camera calibration is necessary before texture mapping when the photogrammetric back-projection is carried out. In contrary to most commercial 3D VR-GIS packages that use only one texture to simulate all facades of a building, results in a "virtual realistic" (VR) city model. In this paper, we adopt photogrammetric techniques that can generate different textures for different faces of a building. By this means multi-face texturing can be apply for each building results in a "photo-realistic" city model.

For the purpose of multi-face texture generation using single photo, the orientation parameters of the camera are required. In this work, it is assumed that polyhedral building models are ready for texture mapping. Thus, the building's roof and foot corners can be utilized as ground control points in space resection for solving the orientation parameters. However, a traditional standard procedure for orientation modeling is tedious. A graphic user interface (GUI) was designed to reflect the orientation modeling result in real-time by the back-projection of the building's wire-frame model on the terrestrial image via the marking and moving of image control points. The operator can visually inspect the consistency between the wire-frame model and the buildings in the terrestrial image. Even a non-expert inexperienced operator can produce facade textures without a long training time.

For the purpose of high production throughput, the buildings are captured in one photo as many as possible provided it can fulfill the required image spatial resolution, e.g. 20cm or 40cm per-pixel. The spatial resolution can be estimated from the image scale and the camera's pixel size. During the texturing mapping stage, the visible faces are first draped with the corresponding texture as determined by photogrammetric back-projection. For occlusion area, it can be composite from different views of image [6], [7]. However some other issues may be raised, such as differences in illumination can cause grey value discontinuity, miss-registration can introduce a ghost effect, and production efficiency can be degraded, and so on.

In this paper, the texture in the non-occluded area is directly created from the visible face. For a partially occluded area, the texture can be generated by filling its neighboring texture using the *mirror* operation. It is assumed that the building texture is repetitive or homogeneous. Since there is no need for multi-view image composition, the number of pictures can be reduced while still maintaining a good textural quality.

Since the quantity of pictures captured for a city model is large, the process of searching for and managing the corresponding pictures for texture production is time consuming. Since the GPS equipment is integrated with the camera, the geographic location can be saved in the header of the picture. When integrated into a GIS environment, the pictures can be managed by their geographic location. We design an emulated GIS interface to mange pictures in a similar way. It is also integrated with the texturing mapping GUI, to assist the operator in their searching for and management of the massive number of terrestrial photos.

2 Methodology

In the following sections, a detailed description regarding camera calibration, multi-face texture mapping, and picture management by GPS-Photo-GIS integration is provided.

2.1 Camera Calibration

A high-resolution DSLR camera is utilized for the experimental investigation, i.e. Nikon D2X with AF-S DX 17-55mm lens. The camera has a lens with a focal length from 17mm to 55mm and a 23.7mm x 15.7mm CMOS sensor. The camera enables a FOV ranging from 79° to 28°, and the resolution is 4,288 x 2,848 pixels, which enables the preservation of rich textural information when distance acquisition is required. The high-resolution wide FOV digital camera is suitable for both taking pictures of a large building from a short distance away or many buildings a long distance away. In such cases, the focal length of the camera is short so that lens distortion will be significant, which has to be considered in the geometrical correction process. In the meantime, due to one picture can contain many buildings the number of pictures for texture generation will be minimized.

For camera calibration, the PhotoModeler [8] is utilized to estimate the lens distortion parameters. Lens distortion includes both radial and decentering parts. The PhotoModeler utilizes a well-designed calibration sheet for camera calibration as shown in Fig.1. It utilizes a series of repetitive pattern for ground control points. At first, four ground control points with known distances are manually measured. The other control points are then automatically matched to increase measurement redundancy. A self-calibration bundle adjustment with additional lens distortion parameters is adopted. Due to an accurate and CPU intensive lens distortion correction model is not necessary for texturing generation, we correct for dominant radial distortion only. A detailed description of the method can be found in [9].

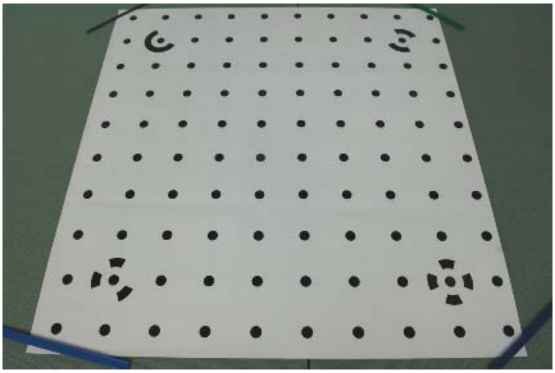


Fig. 1. Camera calibration sheet

2.2 Texture Mapping

In the following sections, the design of the GUI for texture mapping, camera orientation modeling procedure, the multi-face texture generation procedure, and the occlusion compensation procedure are described.

2.2.1 Design of GUI

A graphics user interface for interactive texture mapping based on the Visual Studio .NET platform is developed. The standard OPENGL graphic library is a favorable choice for 3D graphics programming because most commercial graphic cards have hardware implementation that can accelerate the texture rendering. An example of the designed GUI is illustrated in Fig. 2. The GUI has a *3D viewer* to display the 3D building models, to select the facades of interest, and to show the texture mapping results. It also provides a *2D viewer* for displaying terrestrial photos, for marking image control points, and for visual inspecting the buildings wire-frame model on the image. The 2D viewer also allow for another GPS-Photo-GIS integration function, which emulates a GIS environment and can be used to manage the massive number of terrestrial photos.

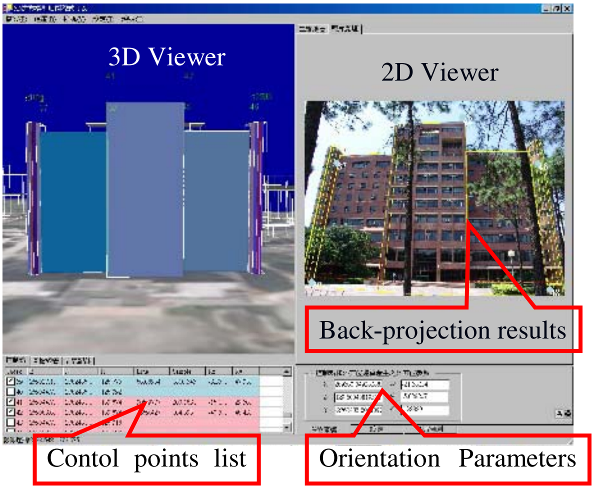


Fig. 2. The designed GUI for texture mapping

In additional, a *tablet frame* is designed to illustrate the corresponding ground control points, list of the image control points and the camera’s orientation parameters.

2.2.2 Control Point Marking and Orientation Modeling

In the 3D viewer, the operator can manually or automatically select more than one facade of interest for texture generation even though occlusions exist. The rooftops and footing corners of the selected facades are labeled and used as candidate ground control points. They are automatically indexed and inserted into the tablet frame. Some of the ground control points are visible in the terrestrial image. The operator can now mark its corresponding image control points from the 2D viewer. After four control points have been marked, the space resection process can be initiated. The orientation results are used for the back-projection of the selected facades and plotting the wire-frame models on the terrestrial image. The operator can then visually inspect them for consistency to confirm the correctness of the orientation modeling. If the orientation result is not correct, the operator needs to move the image control points and the space resection will response in real-time. The orientation modeling is finished when the wire-frame models coincide with the corresponding features in the terrestrial image.

Fig. 3 illustrates two examples of the correct and incorrect orientation modeling results. The left hand column illustrates the texture mapping result, while the right hand column demonstrates the visual inspection procedure. In this demonstration, three facades are selected and four control points are marked. The wire-frames of the selected facades are back-projected onto the terrestrial image based on the orientation modeling result.

2.2.3 Texture Generation and Occlusion Compensation

Once the orientation parameters have been determined, the textures of selected facades can be generated simultaneously. The technique is the same as for orthophoto generation using the photogrammetric back-projection method, i.e. utilize the orientation parameters and the co-linearity equations. However, there may be partial self-occlusion problems with facades. Occlusion detection can be performed by the ZI-Buffer technique similar to the true-orthophoto generation [3].

In this work, it is assumed that building facades contain repetitive patterns or are homogeneous in texture. It is estimated that about 90% of the building models processed hold true to this assumption. Hence, the hidden part can be filled with texture from neighboring parts using the mirror operation, which can be done in autonomous after detection of the major hidden side.

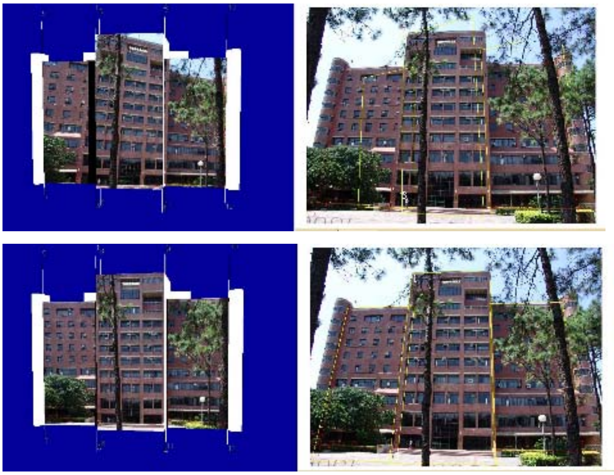


Fig. 3. Demonstration of (upper row) incorrect and (lower row)correct orientation modelling

An example of self-occlusion is demonstrated in Fig. 4. Façade AB is partially occluded by façade CD. The two facades can be projected to the image plane and their distance to the camera's perspective center can be calculated. The distance is recorded in the Z-buffer and an index map is adopted to indicate the hidden pixels. An index map is also used to store the generated texture.

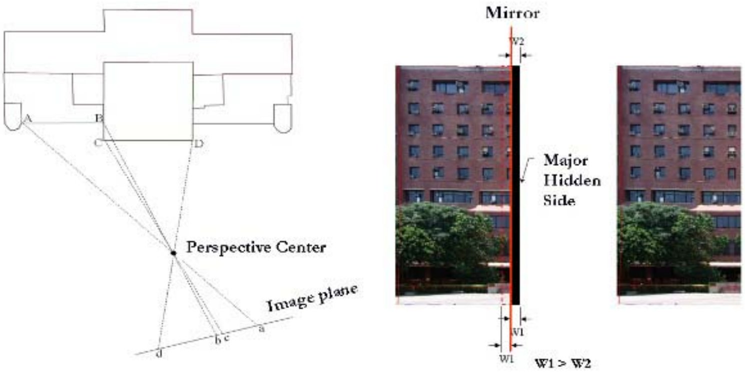


Fig. 4. Illustration of self-occlusion and mirror operation for compensation

As shown in the middle of Fig. 4, the hidden parts are denoted as black color. The major hidden side is detected by searching for black pixels along the four sides. The side containing the largest percentage of black pixels is defined as the major hidden side. In addition, the width of black pixels next to the major hidden side (e.g. w1 and w2 in Fig. 4) is estimated. The location of the mirror can be determined according to the largest one (i.e. w1 due to w1 is greater than w2). The right hand photo in Fig. 4 depicts the texture mirroring result. If one compares the image before and after occlusion compensation, it is demonstrated that the effect of mirror texturing is acceptable if the facade texture has a repetitive or homogeneous pattern. In Fig. 5 the results of multi-face texture mapping by means of the proposed procedure is demonstrated. Except for trees in front of the buildings that are difficult or impossible to remove, the experimental results confirm the fidelity of the proposed scheme. The above results are acceptable and useful for most geo-visualization applications. Moreover, since the number of pictures needed to create the whole building is reduced, the efficiency is significantly improved.

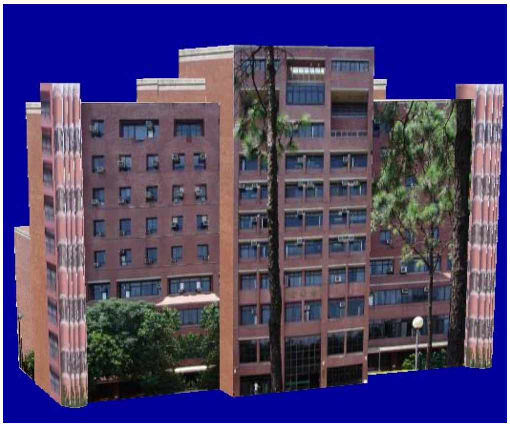


Fig. 5. Example of the multi-faces texture mapping

2.2.4 Picture Management by GPS-Photo-GIS Linking

In the experiments, we integrate the Nikon D2X camera with a Garmin GPS Geko 301. The image was stored in JPEG format with an EXIF header that can store GPS geographic information. In the design of GUI, as shown in Fig. 2, we create a GIS-like environment in the 2D viewer to superimpose an aerial orthophoto and the picture location by geographic coordinates. However, since the orthophoto is in the TWD67 coordinate system (i.e. Taiwan Datum 1967), while the GPS location information is recorded in the WGS84 geographic coordinates (i.e. Lat. / Long.). Before superimposition, the GPS coordinates have to be transferred to the TWD67 coordinate system. Fig. 6 illustrates an example of the overlay results. In the figure, the locations of the pictures are denoted by the yellow triangles. They are linked to the original image in

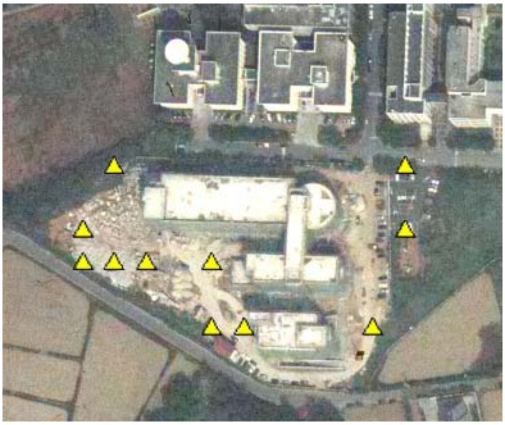


Fig. 6. Example of GPS-Photo-GIS integration for picture management

the hard disk. The operator can thus double-click on the symbol to retrieve and display the corresponding terrestrial photo on the 2D viewer for control point marking. By means of GPS-Photo-GIS integration a large quantity of terrestrial pictures can be managed easily, which is useful and efficient, especially when a city model need to be treated.

3 Case Study

3.1 Camera Calibration

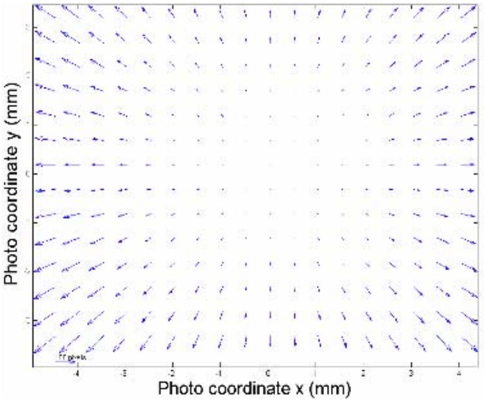
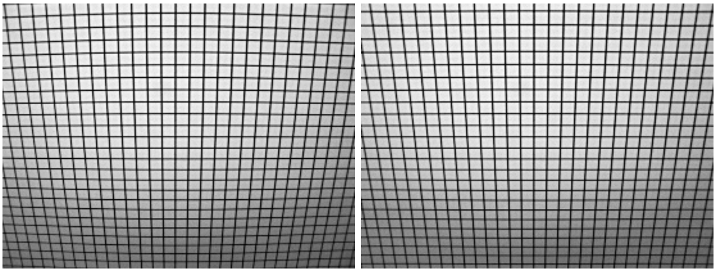


Fig. 7. The total lens distortion vectors

In the experiments, eight different views of photos are acquired using the shortest focal length, i.e. 17 mm. One photo is shown in Fig. 1. The total lens distortion vectors, including radial and decentering distortion, are illustrated in Fig. 7. The maximum lens distortions for radial, decentering and total distortion for a focal length of 17 mm are summarized in Table 1. In Table 1 and Fig. 7, we notice that the total lens distortion is about 91.55 pixels at the edges of the camera frame. The radial lens distortion dominates the total lens distortion. We also adopt the camera calibration result to compare the effect of lens distortion correction. Fig. 8 shows a testing picture with an equal grid. Due to lens distortion the grid appear to bend outward, as shown in Fig. 8 (A). After applying the lens distortion correction procedure, the distorted lines are restored and presented as parallel lines, as shown in Fig. 8 (B). The results demonstrate that the lens distortion is significant and indispensable for texture mapping. The adopted lens distortion correction method is feasible for the creation of a photo-realistic city model.

Table 1. Summary of maximum lens distortion on x-y axis. (Units: Pixels)

	x-axis	y-axis
Radial Lens Distortion	88.45	59.83
Decentering Lens Distortion	3.10	1.62
Total Lens Distortion	91.55	61.46



(A) Before correction (B) After correction

Fig. 8. Comparison of lens distortion correction

3.2 Multi-face Texture Mapping

In this section, we discuss the testing of the 3D building models on NCU. Fig. 9 shows an example of a complex building composed of circular and rectangular shapes. In Fig. 10, 12 visible faces are selected for texture mapping. In Fig. 9, 5 control points, depicted with square symbols, are marked for orientation modeling. The texture mapping result is shown in Fig. 11 for comparison.

In this case, if affine transformation is used for texture mapping. The operator needs to mark four image control points for one face. This means that the operator needs to mark 48 image control points. The man-power needed is significantly more than for the proposed scheme. In additional, the marking of the image control points may introduces a certain degree of measurement errors, so that a gray value discontinuity effect may occur between two consecutive faces when affine transformation is utilized. The described effect does not happen in the proposed approach. The image control points for texture mapping are calculated by the co-linearity equations with the estimated camera orientation parameters. This means that the visual quality is improved.

With this procedure we create a photo-realistic city model of NCU that containing about 300 polyhedrons, as shown in Fig. 12. In total, more than 1,400 terrestrial photos are utilized and about 40 man-hours are spent in texturing mapping. Since the buildings in NCU are separated by some distance taking pictures is not a problem. However, trees or cars in front of the buildings give rise to significant non-self-occlusion problem making it difficult to solve. There is a trade-off between efficiency and realism. On the city scale, the proposed approach fulfills the requirements for producing a photo-realistic city model.

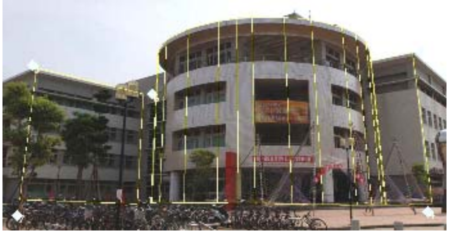


Fig. 9. Terrestrial photo

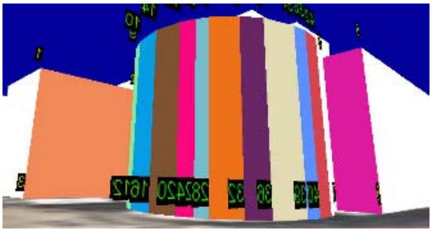


Fig. 10. The selected facades of interest before texture mapping

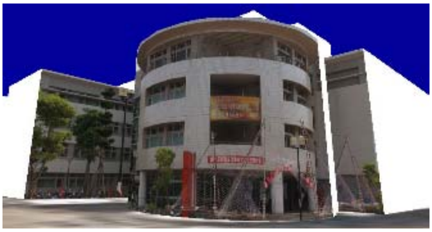


Fig. 11. Multi-faces texture mapping results

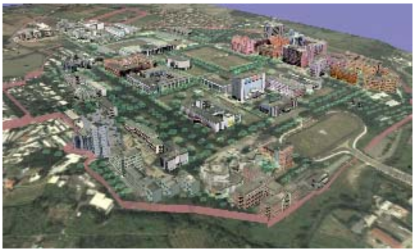


Fig. 12. Photo-realistic city model of NCU

4 Conclusions

In the paper, we propose the use of a high-resolution non-metric wide FOV digital camera for texture generation in photo-realistic city modeling. The adopted camera calibration model is both effective and feasible for texture mapping. The designed GUI interface for control point marking and camera orientation modeling is easy to operate that an expert or experienced operator is not required. The proposed scheme has proven to be efficient due to the following two reasons. The first one is the number of pictures to be process can be reduced. That is because all visible facade texture can be generated from one photo and partially occluded facades can be compensated for using the mirror operation. The fidelity of the generated texture is high provided the facades meet the assumption that they contain repetitive or homogenous patterns. The second reason is that a large quantity of pictures can be treated efficiently by GPS-Photo-GIS integration. Due to the adoption of photogrammetric techniques multi-face texture mapping can be performed, resulting in a photo-realistic city model. The generated city model is appropriate for most geo-visualization applications.

Acknowledgments

This work was partially supported by the NCU-ITRI Union Research and Developing Center under project number NCU-ITRI 930304. It is also partially supported by the National Science Council under project number NSC 93-2211-E-008-026.

References

1. Baillard, C. and A. Zisserman: A Plane Sweep Strategy for the 3D Reconstruction of Buildings from Multiple Images. *IAPRS*, Vol. 33, Part. B2. Amsterdam, Netherlands (2000) 56-62.
2. Brenner, C.: Towards Fully Automatic Generation of City Models. *IAPRS*, Vol. 33, Part. B3, Amsterdam, Netherlands (2000) 84-92.
3. Rau, J. Y., Chen, N.Y. and Chen, L. C.: True Orthophoto Generation of Built-Up Areas Using Multi-View Images. *PE&RS*, Vol. 68, No. 6 (2002) 581-588.
4. Brnner, C., & Haala, N.: Fast Production of Virtual Reality City Models. *IAPRS*, Vol.32, No.4 (1998) 77-84.
5. Haala, N.: On the Refinement of Urban Models by Terrestrial Data Collection. *IAPRS*, Vol.35, Part.B3, Istanbul, Turkey (2004) 564-569.
6. Klinec, D. and Fritsch, D.: Towards Pedestrian Navigation and Orientation. In: Proceedings of the 7th South East Asian Survey Congress, SEASC'03. Hong Kong (2003) On CD-ROM
7. Varshosaz, M.: Occlusion-Free 3D Realistic Modeling of Buildings in Urban Areas. *IAPRS*, Vol.35, Part.B4, Istanbul, Turkey (2004) 437-442.
8. Photomodeler, Eos Systems Inc., Available: http://www.photomodeler.com.
9. Michael R. Bax: Real-Time Lens Distortion Correction: 3D Video Graphics Cards Are Good for More than Games. Stanford Electrical Engineering and Computer Science Research Journal, Spring (2004), Available: http://ieee.stanford.edu/ecj/spring04.html.

Focus + Context Visualization with Animation

Yingcai Wu, Huamin Qu, Hong Zhou, and Ming-Yuen Chan

Department of Computer Science and Engineering
Hong Kong University of Science and Technology
{wuyc, huamin, zhouhong, pazuchan}@cse.ust.hk

Abstract. In this paper, we present some novel animation techniques to help users understand complex structures contained in volumetric data from the medical imaging and scientific simulation fields. Because of the occlusion of 3D objects, these complex structures and their 3D relationships usually cannot be revealed using one image. By our animation techniques, the focus regions, the context regions, and their relationships can be better visualized at the different stages of an animation. We propose an image-centric method which employs layered-depth images (LDIs) to get correct depth cues, and a data-centric method which exploits a novel transfer function fusing technique to guarantee the smooth transition between frames. The experimental results on real volume data demonstrate the advantages of our techniques over traditional image blending and transfer function interpolation methods.

1 Introduction

Volume visualization helps people gain insight into volumetric data using interactive graphics and imaging techniques. The data from real applications such as medical imaging and computational fluid dynamics often contain multiple complex structures. Because of the occlusion of 3D objects, revealing all these structures and presenting their 3D relationships in one image are very challenging. Two widely used techniques to attack the occlusion problem are *direct volume rendering* (DVR) and *focus+context visualization*. By assigning different transparency values to the voxels of the volume data via *transfer functions* (TFs) and then compositing them into one image, DVR can reveal more information than traditional surface-based rendering. Not all structures are interesting to users. Some structures are more important and are often called the *"regions of interest"* or *"focus regions"*, while other structures are less important and only serve as the *"context"*. Various *focus + context* techniques such as illustrative visualization and smart visibility have been developed to reveal the regions of interest (*i.e.*, focus regions) while preserving the context. Nevertheless, all these techniques have their limitations and cannot totally solve the occlusion problem. Usually, when the focus regions are highlighted, the context may be missed or distorted to some extent. On the other hand, the focus may be occluded by the preserved context.

We propose to use animation for focus+context visualization. We first generate some keyframes which can clearly reveal either the regions of interest or the context, and then fill in the gap between each pair of the successive key frames with a sequence of intermediate frames. The spatial relationship between the focus and the context will be revealed in the animation process. Our work is motivated by an interesting psychology

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1293–1302, 2006.

phenomenon - *"bird in cage"'* - also called *afterimage* [1]. It is closely related to the theory of *persistence of vision* which accounts for the illusion of motion in the filming industry. According to the theory, a visual image will persist in a user's brain for a short time after the real image has disappeared from his/her eyes. Therefore, if we quickly show a sequence of the key frames and intermediate frames containing the focus and context, then the focus and context may simultaneously appear in users' brains and the users can figure out their 3D relationship.

Compared with traditional images, *direct volume rendered images* (DVRIs) have some special features. Traditional images usually show objects in the real world so opaque surfaces are often presented, while DVRIs are used to reveal information contained in 3D volume data so multi-layer transparent surfaces are usually displayed. In some DVRIs containing fuzzy structures, there are even no clear surfaces. To generate animations for DVRIs with large transparent areas, depth cues and smooth transitions are especially important for conveying correct information. If we directly apply traditional animation techniques such as image blending to DVRIs, we may not get correct depth cues and some misleading information will be introduced. To address these problems, we propose two animation techniques, *i.e.*, *image-centric method* and *data-centric method*, for DVRIs. In our image-centric method, the intermediate frames between any two successive keyframes are synthesized by image blending. We employ *layered depth images* (LDIs) to provide the depth information of the 3D structures in the keyframes and then composite these LDIs in depth-sorted order to obtain the intermediate images with correct depth cues. The image-centric method provides a practical solution for simple volumetric data and low-end platforms. However, since the operations of the volume rendering are nonlinear, the image-blending method may fail for complex data. Thus, a data centric method is introduced for high-end applications. In our data centric method, the intermediate frames are all direct volume rendered to reveal complex structures and guarantee correct depth cues. The TFs for the intermediate frames are generated by a novel TF fusing technique to guarantee a smooth transition between any two frames. We further develop two animation editing techniques, *i.e.*, *level-of-detail* and *zoom-in/out* to emphasize the focus frames and suppress the context frames in the time domain of the animation.

This paper is organized as follows: After reviewing previous work in Section 2, we give an overview of our animation approaches in Section 3. The image-centric approach for generating animations is presented in Section 4, while the data-centric approach is described in Section 5. In Section 6, we discuss how to emphasize the regions of interest while preserving the context in the time domain of the animation. The experimental results are presented in Section 7. We conclude our work and suggest some future research directions in Section 8.

2 Related Work

Image Blending. Image blending has been extensively studied in image processing [2], computer vision [3], and human-computer interaction [4]. Whitaker et al. [2] proposed an image blending procedure by progressively minimizing the difference of the level sets between two images. Baudisch et al. [4] developed an approach which relies on a vector of blending weights to display overlapping windows simultaneously. More recently, Soille [3] presented a method for image compositing based on mathematical morphology and a marker-controlled segmentation paradigm. Image blending is a classic technique and is still widely used.

Focus + Context Visualization. To effectively visualize salient features (or focus regions) without occlusion while preserving necessary context is a challenging problem in volume visualization. Weiskopf et al. [5] proposed several interactive clipping methods by exploiting the powerful capability of GPUs. However, some spatial relationships of 3D structures are lost when too much context is removed. Viola et al. [6] recently proposed a novel importance-driven approach which is capable of enhancing important features while preserving necessary context by generating cut-away views and ghosted images from volumetric data. It is further used in an interactive system [7], *volumeshop*, for direct volume illustration. Nevertheless, their approaches require pre-segmented volumetric data, which are not always available in practice. Volume illustration, first introduced by Ebert et al. [8], provides an alternative way for focus + context visualization. With *non-photorealistic rendering* (NPR) techniques, volume illustration helps users perceive visualization results by highlighting significant features and subjugating less important details. Lens and distortion is another focus + context technique for volume exploration. Wang et al. [9] developed some interactive GPU-accelerated volume lens to magnify the regions of interest and compress the other volume regions, but at the cost of distorting the data.

Animated Visualization. Lum et al. [10] presented an impressive visualization technique called *Kinetic Visualization* for creating motion along a surface to facilitate the understanding of static transparent 3D shapes. Weiskopf [11] described some relevant psychophysical and physiological findings to demonstrate the important role of color in the perception of motion. Correa et al. [12] proposed a new data traversal approach for volumetric data with motion-controlled transfer functions. Three perceptual properties of motion: flicker, direction, and velocity have been thoroughly examined by Huber et al. [13] in an experimental study. Although there has been growing interest in animated visualization, most previous works either aim at enhancing the perception of 3D shapes and structures of static objects [10, 12] or focus on evaluating some general motion attributes [11,13]. Using animation for focus + context visualization is still an unexplored area.

3 Overview

In this paper, we propose two approaches, *i.e.*, the image-centric and data-centric methods, to create animations for focus + context volume visualization. Our approaches employ an important animation technique called *tweening* (or in-betweening) which can generate a sequence of intermediate frames between two successive key frames so that the first key frame can smoothly change into the second one. After creating a series of key frames that reveal either the focus or the context, our approaches will apply the tweening techniques to the successive key frames to form an animation.

The differences between the image-centric method and the data-centric method lie in three aspects. First, the approaches to generating key frames are different. The image-centric method creates the key frames by surface-based volume rendering with user-specified isovalues, while the data-centric method generates the key frames by DVR with user-specified transfer functions (TFs). Second, the processes of tweening between the key frames are different. The image-centric method achieves the tweening by blending the key frames with different alpha values. The original volume is not required for the tweening process. The data-centric method generates a sequence of intermediate frames by fusing the transfer functions and it requires the original volume. Third, the image-centric method aims at low-end platforms with limited memory and storage space, and low-end applications with simple data. Instead, the data-centric method is used to create higher quality and smoother animations on powerful platforms. The below table summarizes the differences.

	Image-Centric Method	Data-Centric Method
Keyframe Generation	Surface-based volume rendering	Direct volume rendering
Tweening	Blending keyframes	Fusing TFs
Original Volume	Not required for tweening	Required for tweening
Applications	Low-end	High -end

Our system further employs two widely used techniques, level-of-detail and zoom-in/out, to the generated animations to emphasize the important frames (*i.e.*, the focus frames) and suppress the less important frames. They are applied to the animation timeline which controls the display time of each frame in an animation. With the level-of-detail technique, users are able to reduce the number of context keyframes, while the zoom-in/out technique allows users to emphasize the focus frames while suppressing the context frames.

4 Image-Centric Animation

In this section, we present the image-centric method. Compared with the data-centric animation scheme (see Section 5), it is light-weight and better suited for low-end platforms such as smartphones and personal digital assistants (PDA). For example, it can be used in a client-server graphics system. The required key frames (*i.e.*, a series of layered depth images (LDIs)) can be generated in a remote server. Some of those LDIs contain the regions of interest, while the rest have the context. The key frames are then sent to the low-end clients. After receiving the key frames, the clients will fill in the gap between each pair of the successive key frames by blending the key frames with different alpha values. It consumes much less bandwidth, memory, and computation time than the data-centric method.

4.1 Layered Depth Images

The concept of layered depth images (LDIs) was first proposed by Shade et al. [14] for image-based rendering. In our work, the LDIs are used as key frames to provide the depth information of different structures in the original data. The focus region and context region are specified by users as iso-values. Then surface-based volume rendering will be used to generate all key frames within one pass. The LDIs are stored as the result. We cast a ray from each pixel, and save the depth and interpolated color for each intersection point between the ray and the user-specified iso-surfaces.

There are two reasons that we use LDIs as key frames. First, LDIs consume much less memory and bandwidth than original volumes, and can be generated efficiently. Therefore, they are very suitable for low-end platforms. Second, LDIs can guarantee that the blending used in the tweening process (see Section 4.2) and level-of-detail approach (see Section 6) will generate correct depth cues for intermediate frames.

4.2 Tweening Between Key Frames

We employ a simple yet effective technique to fill in the gap between each pair of the successive key frames. The tweening technique creates a sequence of intermediate frames by blending the successive key frames with different alpha values. We denote the image blending as a function $B(P_1,P_2,\alpha_1,\alpha 2)$ where P_1 and P_2 are two successive key frames, and α_1 is an alpha value for P_1 and α_2 is an alpha value for P_2. Thus, the intermediate frames can be viewed as a set of $B(P_1,P_2,\alpha_1,\alpha 2)$ with a linear combination of α_1 and α_2 of the key frames. To get ccorrect depth cues, the layers of LDIs at the similar depth range will be alpha blended first, and then the different layers will be composited into an intermediate frame in depth-sorted order.

5 Data-Centric Animation

The image-centric animation scheme is effective for simple volume data and low-end applications. However, because not all 3D structures can be represented as iso-surfaces, some important information contained in complex volume data may be missed in the keyframes and intermediate frames generated by our method. Moreover, the alpha blending operations may introduce some misleading information [4]. Although we could minimize the amount of incorrect information with LDIs, the existence of misleading or incorrect information is still unavoidable in the blended images in some situations. Therefore, for high-end applications, we propose a data-centric method where all frames are generated by direct volume rendering to provide correct depth cues and eliminate misleading information. Our method applies a novel transfer function fusing technique to generate a sequence of intermediate transfer functions from any two successive keyframe transfer functions, and the animation formed by the images rendered using these transfer functions will guarantee the smooth transitions between frames.

5.1 Similarity-Based Tweening Between Key Frames

TFs in DVR are used to classify different features in volumetric data and determine the information shown in the final images. We assume that multiple key TFs for the key frames have already been generated by users. These TFs either emphasize the focus or the context of the data. Suppose that the TFs for two successive key frames, P_1 and P_2, are TF_1 and TF_2, and the number of intermediate frames between two key frames is N. To tween between these two key frames, one straightforward solution is to linearly interpolate the key TFs to obtain a number of intermediate TFs, which can then be used to generate a sequence of intermediate frames by DVR. For example, the TF for the ith intermediate frame can be computed as $TF_i = \frac{(N-i)}{N} * TF_1 + \frac{i}{N} * TF_2$. However, based on our experiments with real data (See Fig. 1(a)-1(e)), this method cannot generate expected results. This is because that the compositing operation used in volume rendering is nonlinear, and thus the smooth transition between two TFs cannot guarantee the smooth transition between the resulting DVRIs. To solve this problem, we propose a similarity-based tweening technique. We first introduce a similarity metric between two frames and then linearly change the similarity values between the intermediate frames and the key frames. To guarantee a smooth animation, for the ith intermediate frame P_i, the similarity value between P_i and P_1 should be $\frac{(N-i)}{N}$ and the similarity value between P_i and P_2 should be $\frac{i}{N}$. In our data-centric method, we compute a series of intermediate TFs whose generated DVRIs will linearly change their similarity values with the keyframes. To do this, we apply a novel transfer function fusing technique.

5.2 Transfer Function Fusing

Consider the following TF fusing problem: Given two transfer functions TF_1 and TF_2 (called parent transfer functions) and their corresponding DVRIs P_1 and P_2, we are asked to generate a transfer function TF (called child transfer function) whose corresponding DVRI P will have a similarity value v_1 with P_1 and a similarity value v_2 with P_2. We develop a transfer function fusing system to solve this problem based on our previous work [15]. Our system first employs a contour-based similarity metric to compare two direct volume rendered images. Then an energy function is introduced and the transfer function fusing problem is turned into an optimization problem which can be solved using genetic algorithms (GAs).

6 Animation Editing

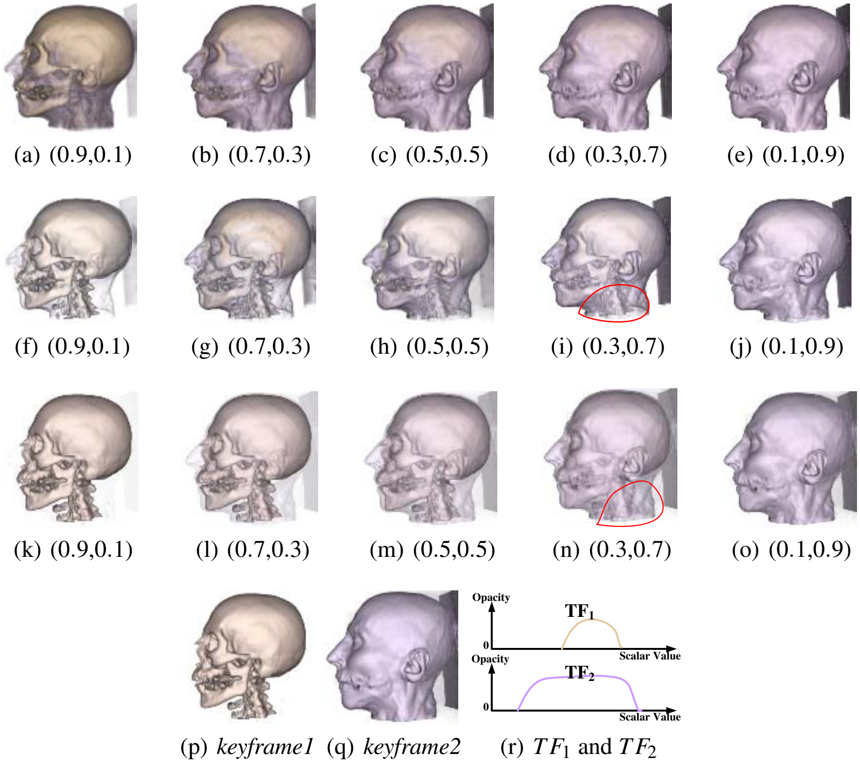
Each key frame used in the animation usually contains only one specific feature except in a few situations where the feature is hard to be separated from other features. If there are many features in the volumetric data, there will be a number of key frames in the animation. Thus, the total number of frames including the key frames and intermediate frames will be too large, which makes it difficult for users to focus on some frames containing the features they want to see. To overcome this problem, we propose two techniques, level-of-detail (LOD) and zoom-in/out, to emphasize the important frames while suppressing the context frames. The proposed techniques are applied on the *timeline*, which controls the display time of each frame.

In our paper, we apply LOD to the timeline to make the exploration easier and more flexible in a visualization process. Users are allowed to select multiple successive key frames and combine them into a new key frame. After that, the system replaces these keyframes with the newly combined key frame for the animation. Tweening will be automatically done with the image-centric or the data-centric method. Users can perform this combination many times until they obtain satisfactory animations. For the image-centric method, the combination of the key frames is done by blending them with the same or different alpha values specified by users. On the other hand, the data-centric method can use the fusing technique for TFs with different expected similarity values to combine the key frames. In our work, we mainly use the LOD technique to reduce the number of the context keyframes.

After the context keyframes are clustered by the LOD technique, we may need to further emphasize the frames in which we are interested. In order to highlight these focus frames, we apply a technique simliar to the fisheye-view approach [16] to the timeline of the animation, which gives the important frames a longer display time and reduces the display time for less important frames.

7 Experiment Results

We tested our system on a Pentium(R) 4 3.2GHz PC with an Nvidia Geforce 6800 Ultra GPU with 256MB RAM. The sampling rate of DVR was two samples per voxel along each ray. The image resolution used in this system was 512 × 512. All volumetric data used for experiments were 8-bit data. For the generation of key frames, the image-centric method is at least n (the number of the required key frames) times faster than the data-centric method, since the image-centric method needs only one pass of surface-based volume rendering to generate all the key frames while the data-centric method



(a) (0.9,0.1) (b) (0.7,0.3) (c) (0.5,0.5) (d) (0.3,0.7) (e) (0.1,0.9)

(f) (0.9,0.1) (g) (0.7,0.3) (h) (0.5,0.5) (i) (0.3,0.7) (j) (0.1,0.9)

(k) (0.9,0.1) (l) (0.7,0.3) (m) (0.5,0.5) (n) (0.3,0.7) (o) (0.1,0.9)

(p) *keyframe1* (q) *keyframe2* (r) TF_1 and TF_2

Fig. 1. Tweening between key frames: (p) and (q) are the key frames; (a)-(e) intermediate frames created by linearly interpolating TFs of the key frames with different (α, β); (f)-(j) intermediate frames created by fusing the key frames with different (v_1, v_2); (k)-(o) intermediate frames generated by blending the key frames with different (α_1, α_2); (r) TFs of the key frames

requires n passes of DVR to create them. For the tweening, the data-centric method required around 40 seconds to fuse two TFs, while the image-centric method needed only 0.147 seconds to blend two frames.

Fig. 1 shows the differences of the tweening done by different methods. Fig. 1(p) and 1(q) are the key frames indicated as *keyframe1* and *keyframe2*, and Fig. 1(a)-1(e) were generated by linearly interpolating the TFs of the key frames ($TF = \alpha * TF_1 + \beta * TF_2$, where α and β are shown below the corresponding figure, and TF_1 and TF_2 are shown in Fig. 1(r)). Fig. 1(a)-1(e) fail to form a smooth transition from *keyframe1* to *keyframe2*. The change from Fig. 1(a) to 1(b) is too abrupt, and Fig. 1(b)-1(e) are almost the same as the *keyframe2* (Fig. 1(p)). Oppositely, Fig. 1(f)-1(j) generated by our data-centric method and Fig. 1(k)-1(o) created by our image-centric method have a smoother transition between the successive key frames. Additionally, the tweening (Fig. 1(f)-1(j)) done by the data-centric method is the best among all these methods. Fig. 1(k) and 1(o) make the morphing, generated by the image-centric method, a bit abrupt and not as good as that created by the data-centric method, since they make the transition from *keyframe1* to Fig. 1(l) and the transition from Fig. 1(n) to *keyframe2* rough. Moreover, the images created by the data-centric method have richer details and provide better depth cues than those created by the image-centric method (see the regions selected by the red curves in Fig. 1(i) and 1(n)).

The experiment on a CT carp dataset ($256 \times 256 \times 512$) was conducted to demonstrate the effectiveness of the LOD technique used in our system. There are four key frames (Fig. 2(a)-2(d)) generated by the image-centric method, and three key frames (Fig. 2(e)-2(g) generated by the data-centric method. The LOD technique was applied to reduce the number of the context keyframes. Fig. 2(h) was generated by fusing Fig. 2(f), 2(g), and 2(e) with the same expected similarity value ($v = 0.5$), while Fig. 2(i) was generated by compositing Fig. 2(a)-2(d)) in depth-sorted order. Both Fig. 2(h) and

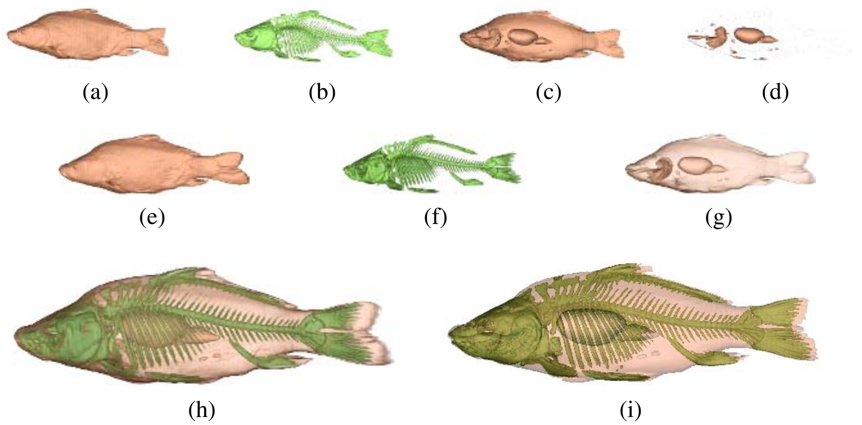


Fig. 2. Key frames: (a)-(d) key frames (LDIs) generated using the image-centric method; (e)-(g) key frames generated with the data-centric method; (h) image generated by fusing (f) with (g), and then with (e); (h) image generated by compositing the LDI (a)-(d) in depth-sorted order

2(i) can correctly provide depth cues. This experiment shows that both the blending and fusing methods are effective for simple volumetric data like the CT carp data.

Finally, our approaches were applied to an MRI head dataset ($256 \times 256 \times 256$). Fig. 3(a)-3(c) are LDIs generated by the image-centric method within one pass of surface-based volume rendering, while Fig. 3(d)-3(f) were generated separately using traditional DVR with manually-created TFs. Obviously, Fig. 3(b) and 3(c) are sharper and clearer than Fig. 3(e) and 3(f). However, Fig. 3(e)-3(f) may better reflect the fuzzy and noisy nature of the original data.

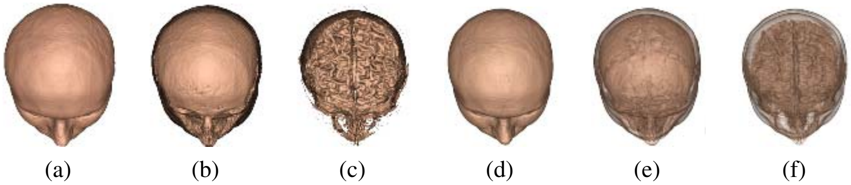


Fig. 3. Key frames: (a)-(c) are LDIs generated using the image-centric method; (d)-(f) are generated using the data-centric method

8 Conclusions and Future Work

In this paper, we have developed two animation techniques - the image-centric and data-centric methods - for focus + context visualization. Our animation techniques allow users to better visualize the focus and the context regions in volume data and reveal their 3D relationship. Layered depth images were exploited in the image-centric method to provide correct depth cues. A novel transfer function fusing method was employed in the data centric method to achieve smooth transitions between frames. The experiment results on real volume data demonstrated the advantages of our methods over traditional image blending and transfer function interpolation approaches. Our animation techniques can help users better understand their data and are very suitable for presentation, education, and data exploration.

In the future, we plan to improve the animation quality of the image-centric method by investigating more advanced blending schemes, and the tweening speed for the data-centric method by exploiting graphics hardware. We also want to conduct a user study to thoroughly evaluate the effectiveness of the animation techniques for focus + context volume visualization and fine tune the parameters used in our animation system.

Acknowledgements

This work was supported by RGC grant CERG 618705 and HKUST grant DAG 04/05 EG02. We would like to thank the anonymous reviewers for their helpful comments, and thank Ka-Kei Chung for his help with the experimental results.

References

1. Robinson, W.S.: Understanding Phenomenal Consciousness. Cambridge University Press, Cambridge, U.K (2004)
2. Whitaker, R.T.: A level-set approach to image blending. IEEE Transactions on Image Processing **9**(11) (2000) 1849–1861
3. Soille, P.: Morphological image compositing. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(5) (2006) 673–683
4. Baudisch, P., Gutwin, C.: Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending. In: Proceedings of the SIGCHI conference on Human factors in computing systems. (2004) 367–374
5. Weiskopf, D., Engel, K., Ertl, T.: Interactive clipping techniques for texture-based volume visualization and volume shading. IEEE Transactions on Visualization and Computer Graphics **9**(3) (2003) 298–312
6. Viola, I., Kanitsar, A., Gröller, M.E.: Importance-driven feature enhancement in volume visualization. IEEE Transactions on Visualization and Computer Graphics **11**(4) (2005) 408–418
7. Bruckner, S., Gröller, M.E.: Volumeshop: An interactive system for direct volume illustration. In: Proceedings of IEEE Visualization 2005. (2005) 671–678
8. Rheingans, P., Ebert, D.: Volume illustration: Nonphotorealistic rendering of volume models. IEEE Transactions on Visualization and Computer Graphics **7**(3) (2001) 253–264
9. Wang, L., Zhao, Y., Mueller, K., Kaufman, A.E.: The magic volume lens: An interactive focus+context technique for volume rendering. In: Proceedings of IEEE Visualization 2005. (2005) 367–374
10. Lum, E.B., Stompel, A., Ma, K.L.: Using motion to illustrate static 3d shape–kinetic visualization. IEEE Transactions on Visualization and Computer Graphics **9**(2) (2003) 115–126
11. Weiskopf, D.: On the role of color in the perception of motion in animated visualizations. In: Proceedings of IEEE Visualization 2004. (2004) 305–312
12. Correa, C.D., Silver, D.: Dataset traversal with motion-controlled transfer functions. In: Proceedings of IEEE Visualization 2005. (2005) 359–366
13. Huber, D.E., Healey, C.G.: Visualizing data with motion. In: Proceedings of IEEE Visualization 2005. (2005) 527–534
14. Shade, J., Gortler, S., wei He, L., Szeliski, R.: Layered depth images. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques. (1998) 231–242
15. Wu, Y., Qu, H., Zhou, H., Chan, M.Y.: Fusing features in direct volume rendered images. In: To appear in the proceedings of the second international symposium on visual computing. (2006)
16. Furnas, G.W.: Generalized fisheye views. In: Proceedings of the SIGCHI conference on Human factors in computing systems. (1986) 16–23
17. S, T., M, H., RA, A.: Human perception of structure from motion. Vision Research **31** (1991) 59–75
18. Lagasse, P., Goldman, L., Hobson, A., Norton, S.R., eds.: The Columbia Encyclopedia. Sixth edn. Columbia University, U.S.A (2001)

Exploiting Spatial-temporal Coherence in the Construction of Multiple Perspective Videos

Mau-Tsuen Yang, Shih-Yu Huang, Kuo-Hua Lo, Wen-Kai Tai,
and Cheng-Chin Chiang

Department of Computer Science & Information Engineering
National Dong-Hua University, Taiwan
mtyang@mail.ndhu.edu.tw

Abstract. We implement an image-based system that constructs multiple perspective videos with active objects using an off-the-shelf PC and digital video camcorders without the need of advanced calibration and 3D model reconstruction. Especially, temporal coherence is exploited to speedup the correspondence search. Spatial coherence is exploited for occlusion recovery to improve the quality of the generated images. The proposed system can automatically establish dense correspondence and construct virtual views on-line according to user instructions.

Keywords: Active object movie, Multiple perspective video, 3D video, View morphing, Occlusion recovery, Spatial-temporal coherence.

1 Introduction

To create an object movie with active objects, called a *multiple perspective video* or *3D video*, an image sequence, instead of a single image, must be stored for each viewing direction to record movement of the objects. The set of image sequences, each captured at separate perspectives but all focused on the same target object, can be analyzed to generate a virtual image sequence at arbitrary viewing direction. Several image-based interpolation techniques have been proposed to generate intermediate views of static scenes with fixed objects [1,10,14]. Recently, some researches have been conducted on the analysis of dynamic scenes with non-rigid or moving objects [5,8,13] with the help of 3D model reconstruction and re-projection.

The spatial-temporal view interpolation [13] was proposed for interpolating views of a non-rigid object across both space and time. They analyzed captured image sequences to compute 3D voxel models of the scene at each time instant and to generate a scene flow across time in which the movement of each voxel is described. In addition, Matsuyama et al. [8] reconstructed 3D shape based on silhouette intersection using a PC cluster. Kitahara et al. [5] represented a 3D object with a set of planes based on the observer's viewing position. Because these methods were based on explicit 3D geometry, the freedom of observer's position is guaranteed. However, not only high-end video cameras and special synchronization hardware were required, but set-up and calibration process was also tedious.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1303 – 1313, 2006.

Instead of relying on explicit 3D model, it is also possible to implement multiple perspective video using image based approach in that implicit 3D model such as correspondence can be utilized. Huang et al. [3] proposed a method to generate arbitrary intermediate video from two videos captured at separate viewing positions. Their interpolation algorithm is based on the disparity based view morphing. Stereo video can be generated, compressed, and displayed on-line. However, the same interpolation algorithm was repeated at each time instant, and temporal coherence was not considered at all. As a result, the computational intensive correspondence search slowed down the speed of the system.

Another interesting problem in multiple perspective videos is occlusion that can degrade the quality of virtual view significantly. Occlusion occurs when some parts of objects can be seen in one perspective, but may disappear in other perspectives. In an object movie, occlusion usually occurs on the left/right boundary of an object. Searching for corresponding points in the occluded area is very difficult if not impossible. However, without corresponding pairs in these areas, the left/right edges of objects in a virtual view will become blurred.

Speed and quality are two critical issues for realizing the potential power of interactive multiple perspective videos. In this paper, temporal coherency is exploited to speedup the correspondence search. Occlusion is handled properly based on spatial coherence to improve the quality of the generated images. The input image sequences can be captured using off-the-shelf DV camcorders without the need for advanced calibration. A sequence of virtual views can be generated and displayed on-line using an inexpensive PC according to the demands of the user. Fig. 1 demonstrates the basic ideas and the flow chart for the proposed system.

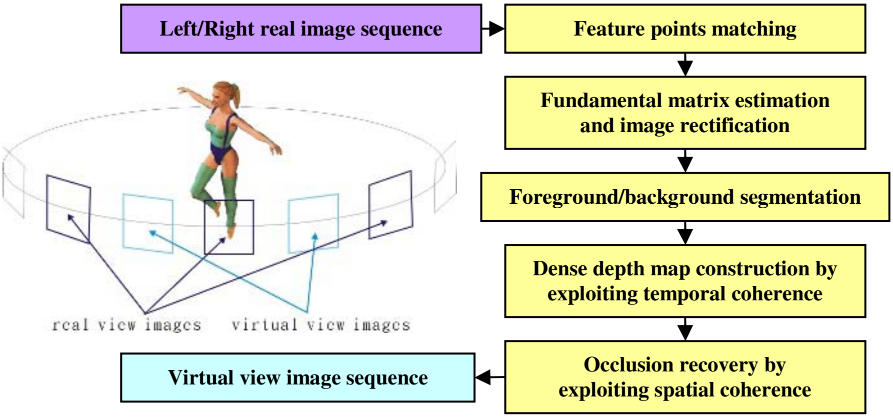

Left/Right real image sequence
Feature points matching
Fundamental matrix estimation and image rectification
Foreground/background segmentation
Dense depth map construction by exploiting temporal coherence
Occlusion recovery by exploiting spatial coherence
Virtual view image sequence
real view images
virtual view images

Fig. 1. Concepts and flowchart of the construction of a multiple perspective video

The remaining parts of the paper are organized as follows. Section 2 explains feature point extraction and corresponding pair matching. Section 3 presents fundamental matrix estimation and image pair rectification. Section 4 describes the foreground/background segmentation. Section 5 utilizes epipolar constraints to efficiently construct a dense depth map. Section 6 exploits temporal coherence to speedup the correspondence search. Section 7 exploits spatial coherence for occlusion recovery to improve the quality of the depth map. Section 8 demonstrates the experimental results. Section 9 presents the conclusions and discussions.

2 Corresponding Feature Point Matching

First, several image sequences are captured using DV camcorders that are located at different viewing positions but focusing at the same target object in the center. To construct virtual views, the feature point correspondence between each neighboring pair of captured image sequences must be established. The correspondence problem is a very challenging problem. Instead of finding corresponding feature points manually or semi-automatically, we implement an efficient block matching method to automatically establish the correspondence. A few salient feature points in both images are found using the Susan edge detector [12]. Typically detected feature points are points with significant changes in brightness in the neighborhood such as the corners of an object. For each feature point on the left (right) image, the corresponding feature point on the right (left) image is searched by comparing the neighborhood similarities calculated based on a *Normalized Correlation* (NC) score defined as follows:

$$\begin{aligned} Score(p_1,p_2) &= \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}[I_1(x+i,y+j)-\overline{I_1(x,y)}]\times[I_2(s+i,t+j)-\overline{I_2(s,t)}]}{u} \\ u &= \sqrt{\sum_{i=-n}^{n}\sum_{j=-m}^{m}[I_1(x+i,y+j)-\overline{I_1(x,y)}]^2\sum_{i=-n}^{n}\sum_{j=-m}^{m}[I_2(s+i,t+j)-\overline{I_2(s,t)}]^2} \end{aligned} \qquad (1)$$

where p_1 represents a pixel (x,y) in the left image I_1, p_2 represents a pixel (s,t) in the right image I_2, $I(x,y)$ represents the brightness intensity of a pixel (x,y) in the image I, and the size of the comparison window is $(2m+1)x(2n+1)$. Point p_2 with the maximum correlation score is selected as the matching candidate for p_1. The results from this step will be a set of corresponding feature point pairs.

3 Fundamental Matrix Estimation and Image Rectification

Epipolar geometry [11] is an effective means to speedup the search for corresponding points. Suppose a point P in 3D space is projected to p on the image plane I_1 with focus C_1 and to point p' on image plane I_2 with focus C_2. The plane contains point p, C_1, and C_2 is called epipolar plane. The epipolar plane intersects the image plane I_1 (I_2) on a straight line L_1 (L_2). The lines L_1 and L_2 are called epipolar lines. For each feature point p on I_1 (I_2), its corresponding point on I_2 (I_1) must fall on the epipolar line L_2 (L_1). The projected point p on I_1 (I_2) and its corresponding epipolar line L_2 (L_1) on I_2 (I_1) are related by a 3x3 matrix F, called fundamental matrix. Their relationships are $Fp=L_2$ and $F^Tp'=L_1$.

The fundamental matrix F contains both intrinsic parameters (such as focal length) and extrinsic parameters (such as relative translation and rotation) of an image pair. If the fundamental matrix is known, then the correspondence search algorithm only needs to search through a 1-D epipolar line instead of through a 2-D search window. Thus, the computation time can be significantly reduced.

A fundamental matrix can be estimated using only 8 corresponding pairs [6]. However, the resulting estimation can be seriously degraded by a single outlier in the input corresponding pairs. The estimation accuracy can be improved by providing more reliable corresponding pairs. The redundancy of corresponding pairs can be utilized to minimize the estimation error. Fortunately, it is assumed that the two acquiring cameras do not move during the capturing period, making the fundamental matrix between them fixed across time. Thus, all of the corresponding pairs obtained at different time instants can be fused together to estimate the fundamental matrix. At time instant t_i, a record is maintained for selecting the top 30 corresponding pairs with the highest correlation scores from all correspondence pairs obtained from time t_0 to t_i. These best corresponding pairs are used to estimate a fundamental matrix using a non-linear minimization method [4,7]. The estimated fundamental matrix is then used to obtain the epipolar lines at time instant t_{i+1}. This iterative process is performed repeatedly at each time instant. As a result, the quality of the top 30 corresponding points becomes more reliable and the accuracy of the estimated fundamental matrix is improved across time as well. In case the estimation error is too large due to emerging outliers, the process is reset and a new set of corresponding points is collected.

With the fundamental matrix an image pair can be rectified to speedup the stereo matching. The purpose of the rectification is to transform the images according to their fundamental matrix so that their epipolar lines are aligned horizontally. In this paper, we apply the polar rectification method [9] so that the matching ambiguity can be reduced to half epipolar lines.

4 Foreground/Background Segmentation

Since we are usually interested only in the foreground objects in an object movie, the foreground and background should be separated so that the dense computation in the next step can focus on the foreground objects. Every pixel in an image can be classified as foreground or background according to its color information. A static background is assumed in our application and the color information for the background can be learned in the startup phase. The startup phase contains N frames in which no foreground object is present. In these fames, the mean $\mu_i(x,y)$ and variance $\sigma_i^2(x,y)$ of each color component $i \in \{R,G,B\}$ in each pixel (x,y) are measured. In the following frames, a distance $d_t(x,y)$ for the pixel (x,y) at time instant t is defined using the following equation to evaluate its likeliness as a foreground pixel:

$$d_t(x,y) = \sqrt{\sum_{i \in \{R,G,B\}} \frac{(I_{i,t}(x,y) - \mu_i(x,y))^2}{\sigma_i^2(x,y)}} \tag{2}$$

where $I_{i,t}(x,y)$ is the value of i-th color component of the pixel (x,y). The background region in the left frame B_l and foreground region in the left frame F_l can be segmented using a threshold T.

$$\begin{cases} B_{l,t} = \{(x,y) \mid d_{l,t}(x,y) < T\} \\ F_{l,t} = \{(x,y) \mid d_{l,t}(x,y) \geq T\} \end{cases} \tag{3}$$

The background (foreground) region in the right frame $B_r(F_r)$ can be defined similarly. After each pixel is classified as foreground or background, a morphological closing operating, i.e. a 3x3 dilation followed by a 3x3 erosion, is performed to fill-in the noisy holes in the foreground region.

5 Dense Depth Map Construction

A dense depth map is constructed using a correspondence search process that is similar to the block matching method discussed in Section 2. However, several differences exist. First, instead of matching feature points, we need to find a corresponding point for every foreground pixel. To reduce the computation complexity, we use a *sum-of-absolute-difference* (SAD) score to replace the NC score.

$$Score(p_1, p_2) = \sum_{i=-n}^{n} \sum_{j=-m}^{m} \left| I_1(x+i, y+j) - I_2(s+i, t+j) \right| \tag{4}$$

Second, since the image has been rectified according to the estimated fundamental matrix, the search window can be limited to a horizontal stripe that is around its epipolar line. By reducing the search area from two dimensions to one dimension, the computation time can be significantly reduced and the quality of corresponding pairs can be improved. In addition, an analysis of the SAD score distribution can help to identify uncertainties. A nearly flat SAD distribution represents a smooth area without much texture, while a SAD distribution with several similar minima indicates areas with repetitive texture. Let SAD_1 be the minimum SAD score that locates at pixel p_r, and SAD_2 be the second lowest SAD score that locates at pixel p'_r in the search window. We define the uncertainty of a corresponding pair $(p_l \rightarrow p_r)$ as $U = SAD_2 / SAD_1$. The corresponding pair $(p_l \rightarrow p_r)$ is accepted only if SAD_1 is lower than a predefined threshold and U is higher than another threshold.

Third, after two correspondence sets including left-to-right and right-to-left mappings are established, two sets of correspondence (left-to-right and right-to-left) are joined together to obtain a single correspondence set. However, the joining set might contain several one-to-many of many-to-one mappings due to occlusion or mismatch. This can be corrected by a left/right consistency check. For each one-to-many (or many-to-one) mapping, we try to replace it using one-to-one mapping that has the lowest SAD score. The final correspondence set will be a single set containing one-to-one dense mappings. With these modifications, a dense depth map can be constructed more reliably and efficiently. Nevertheless, incorrect corresponding pairs and depth estimations are inevitable using an automatic search method.

6 Exploiting Temporal Coherence for Speedup

The construction time of the dense depth map can be further reduced by exploiting the temporal coherence. In an object movie with real-time frame rate, objects in consecutive frames usually stay fixed or move very slowly. Temporal coherence can be exploited to speedup the search of corresponding points. Suppose $P_{l,t}$ ($P_{r,t}$) is a pixel located on a frame captured in left (right) images at time instant *t*, We represent a one-to-one pair of corresponding points at time instant *t* by $(P_{l,t} \leftrightarrow P_{r,t})$, and the set of all corresponding pairs at time instant *t* by CR_t. Inside an image frame at time instant *t*, only certain points require a new search for the corresponding points. Other points can inherit their corresponding points from the previous correspondence set CR_{t-1} at time instant *t-1*. The regions that require a new correspondence search can be categorized to three cases, analyzed as follows:

First, changing region *S* contains pixels with significant changes in brightness. Suppose $I_{l,t}(x,y)$ ($I_{r,t}(x,y)$) represents the intensity of a pixel *(x,y)* located on left (right) image at time instant *t*, the changing region *S* can be detected using a threshold.

$$\begin{cases} S_{l,t} = \{(x,y) \mid \left| I_{l,t}(x,y) - I_{l,t-1}(x,y) \right| > T\} \\ S_{r,t} = \{(x,y) \mid \left| I_{r,t}(x,y) - I_{r,t-1}(x,y) \right| > T\} \end{cases} \tag{5}$$

Second, missing region *M* contains every pixel at time instant *t* that does not have any corresponding point at time instant *t-1*. The missing region *M* is defined using the following equation:

$$\begin{cases} M_{l,t} = \{(x,y) \mid ((x,y) \leftrightarrow P_{r,t-1}) \notin CR_{t-1}\} \\ M_{r,t} = \{(x,y) \mid (P_{l,t-1} \leftrightarrow (x,y)) \notin CR_{t-1}\} \end{cases} \tag{6}$$

Third, invalid region *V* contains every pixel in the frame at time instant *t* that does have corresponding point at time instant *t-1*, but the corresponding point falls inside the changing region *S* or the background region *B* at time instant *t*. The invalid region *V* can be identified using the following equation:

$$\begin{cases} V_{l,t} = \{(x,y) \mid ((x,y) \leftrightarrow P_{r,t-1}) \in CR_{t-1} \ and \ (P_{r,t-1} \in S_{r,t} \ or \ P_{r,t-1} \in B_{r,t})\} \\ V_{r,t} = \{(x,y) \mid (P_{l,t-1} \leftrightarrow (x,y)) \in CR_{t-1} \ and \ (P_{l,t-1} \in S_{l,t} \ or \ P_{l,t-1} \in B_{l,t})\} \end{cases} \tag{7}$$

Re-computing region *R* is then defined as the union of changing region *S*, missing region *M*, and invalid region *V*.

$$\begin{cases} R_{l,t} = S_{l,t} \cup M_{l,t} \cup V_{l,t} \\ R_{r,t} = S_{r,t} \cup M_{r,t} \cup V_{r,t} \end{cases} \tag{8}$$

For each pixel in the left (right) image inside the re-compute region R_l (R_r) a corresponding point in the right (left) image must be searched. For each pixel that is not inside the re-compute region, its corresponding point can be propagated from the last consecutive frames. As a result, the time spent on correspondence search can be reduced greatly. Fig. 2 demonstrates these three cases using a virtual image sequence that contains a small teapot moving horizontally across a static big teapot.

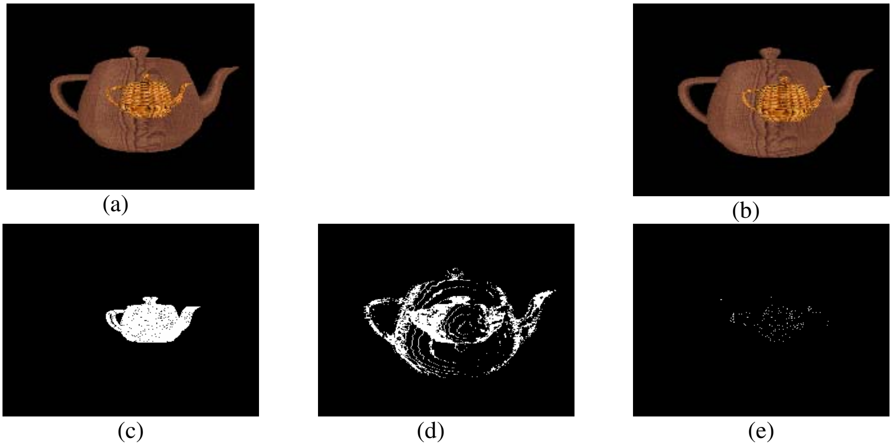


Fig. 2. Exploit temporal coherence to speedup correspondence search. (a) Left image at time *t-1* (b) Left image at time *t* (c) Changing area $S_{l,t}$ (d) Missing area $M_{l,t}$ (e) Invalid area $V_{l,t.}$

7 Exploiting Spatial Coherence for Occlusion Recovery

The quality of the final generated virtual images depends on the quality of the depth map. After the dense depth construction, there will still be some holes in the depth map due to high SAD or low left-right consistency. The quality of the depth map can be further improved by filling these holes properly. In order to preserve the object edges in the process of hole filling, each hole should be filled along scan lines parallel to the primary edge axis of the hole. A primary edge axis for each hole in the depth map can be computed as the majority of the orientation of the corresponding edges in the edge map. An edge map can be constructed by convolving the original image with a Prewitt operator [11]. Furthermore, each hole should be filled using different strategy based on the causes of the hole.

First, in the smooth areas that lack texture, the correspondence search usually fails because of ambiguity, resulting in holes in the depth map. For each scan line in a hole, suppose d_1 (d_2) is the depth value on the left (right) boundary of the line, the depth of any pixel along the scan line can be linearly interpolated using d_1 and d_2.

$$d(s) = d_2 \times \frac{s}{K} + d_1 \times (1 - \frac{s}{K}) \tag{9}$$

where K is the distance between the left and the right boundary and s is the distance between the estimated pixel and the left boundary.

Second, closer objects in the scene might occlude objects that are further away. Some areas appearing in the left (right) image might completely disappear in the right (left) image. The correspondence search will fail again in these areas, causing holes in the depth map. Each hole in this case contains two areas belonging to either closer object or further object respectively. Depth discontinuity should be kept around the occlusion boundary. Instead of using a linear interpolation, a critical point with maximum magnitude in the edge map is found along each scan line. The depth value of any pixel located on the left (right) of the critical point can copy the depth value on the left (right) boundary of the scan line.

$$d(s)=\begin{cases} d_1, & \text{if } s \text{ locates on the left of the critical point} \\ d_2, & \text{if } s \text{ locates on the right of the critical point} \end{cases} \tag{10}$$

Third, certain parts of an object might be occluded by itself, called self-occlusion. In other words, the left (right) boundary of an object that can be seen by left (right) camera might not be observed by the right (left) camera. In this case, the depth along each scan line in the hole should be replaced by the depth of the foreground object, i.e., $d(s)=min(d_1,d_2)$.

Fig. 3 demonstrates these three cases using a virtual image sequence containing a big teapot occluded by a small teapot. Scan lines for hole filling are set to be horizontal in this example for illustration purpose. As a final noise filter, a depth consistency check in the neighborhood is applied to remove outliers. After these refinements, all the empty holes in the depth map are filled, and the depth for each pixel in the foreground region is determined.

After a reliable and dense depth map is established, an arbitrary virtual image is interpolated in any desired viewing direction. First, the left and right images (I_1 and I_2) are warped such that the optical axes of both images (I_1' and I_2') are perpendicular to the transition baseline. Second, a morphing function is applied to the warped images to obtain a desired transition view. The morphing function used here is called disparity based view morphing [3] which is a non-linear mapping based on the depth value of the corresponding pair. Finally, the transition view (I_s') is warped back to the correct view (I_s) with proper viewing direction and position. Holes in the final view are eliminated by interpolating the values of horizontally adjacent pixels.

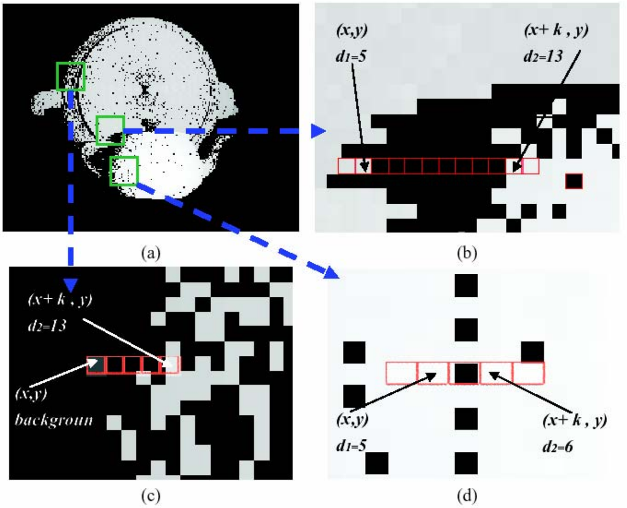


Fig. 3. Three types of holes in depth map. (a) The depth map (b) Further object occluded by closer object (c) Self-occlusion area (d) Smooth area.

8 Experimental Results

We demonstrate the constructing results of two multiple perspective videos containing a real person and a cartoon character respective. The movie lasts for 30 seconds (600 frames) and the acting character rotates his head, shakes his hands, and moves his body throughout this period. The image resolution is 360x240 in a 24-bit color format. Two cameras captured the left and right image sequences in different viewing directions (around 30 degree apart). The proposed system generates and displays a virtual image sequence from the center viewing direction providing left and right image sequences as inputs. The execution speed is around 7 frames per second on a standard PC with a Pentium IV 2.0 GHz CPU. One additional camera captured a real image sequence at center viewing position for comparison purpose. Fig. 4 shows the left, right, virtual center, and real center image sequences.

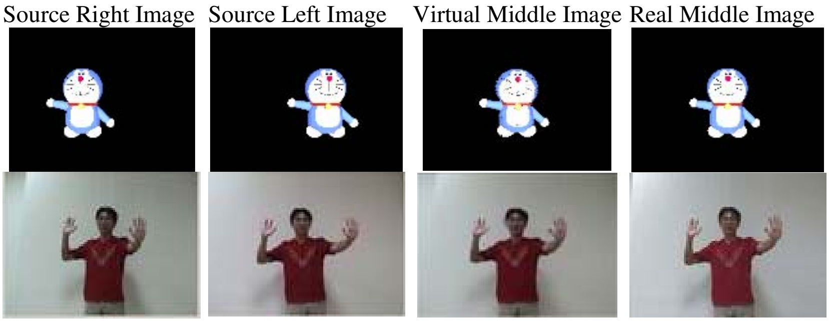


Fig. 4. Screenshots from two multiple perspective videos

The virtual and real image sequences look very similar during the playback. With close examination on frozen images, we notice that two kinds of conditions cause blurred virtual images in this example. The first condition occurs when some parts of acting character move too fast (e.g. shaking hands). Fast movement blurs the capture images so that feature points can not be extracted precisely in blurred area. Thus, the quality of the generated virtual images degrades. The second condition occurs in areas with incorrect corresponding pairs. Incorrect corresponding pairs cause displacement in the interpolation process and also reduce the fundamental matrix estimation accuracy. Generally, the quality of the virtual images in the ending part is better than the quality of the virtual images in the beginning part. The reason is that the estimated fundamental matrix might not be accurate in the beginning. As time goes by, more reliable corresponding pairs become available, so the estimated fundamental matrix is much more stable in the latter part of the sequence.

Fig. 5(a) demonstrates the speedup of the processing time for each frame to construct and refine the depth map. The curve with asterisk marks (*) represents the processing time without considering temporal coherence and occlusion recovery. The curve with plus marks (+) represents the processing time considering only temporal coherency, while the curve with circle marks (。) represents the processing time considering only occlusion recovery. The curve with square marks (□) represents the

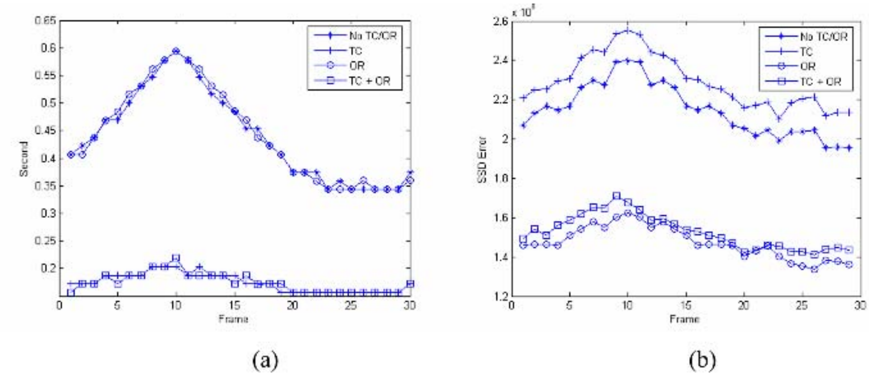


Fig. 5. The comparison of the (a) processing time and (b) quality of the virtual images

processing time considering both temporal coherence and occlusion recovery. It can be noted that the exploitation of temporal coherence not only speeds-up the process by a factor of 3, but also decrease the variance in the processing time across frames. This means that the virtual images can be generated much more frequently and smoothly using the proposed method. It can also be observed that the time spent on occlusion recovery is relatively small due to each hole can be filled with a one pass scan. Fig. 5(b) compares the quality of the generated virtual images through the whole image sequence. The quality is measured using the sum-of-square-difference (SSD) over every pixel between virtual and real image frames. Similarly, four curves with different marks represent the SSD for four different cases. It can be noted that the occlusion recovery improves the quality while the temporal coherence degrades the quality of the generated images. It can also be observed that the quality of the virtual view generally improves across time except around frame 50-60 where the target waves his hands very fast.

9 Conclusions and Discussions

We implement a system that automatically constructs a multiple perspective video from a few image sequences captured at different perspectives. Especially, temporal coherence is analyzed and exploited to speedup the dense correspondence matching between each pair of images. Several strategies are proposed to properly fill holes in the depth map caused by textureless, occluded or self-occluded regions. Only simple equipment (off-the-shelf PC and DVs) is used in the proposed system. Image sequences are captured at separate viewing positions as the input data. The proposed system analyzes the captured image sequences and finds the correspondence between them. At playback time, a virtual image sequence is generated dynamically at arbitrary viewing directions so that a user can change his viewing directions on-line.

Acknowledgements

This research was supported in part by the National Science Council of Taiwan.

References

1. Chen, S. & Williams, L.: View Interpolation for Image Synthesis. *Proceedings of SIGGRAPH*, Pages 279-288 (1993)
2. Gonzalez, R. & Woods, R. *Digital Image Processing*. Prentice Hall (2002)
3. Huang, H., Kao, C. & Hung, Y.: Generation of Multi-Viewpoint Video from Stereoscopic Video. *IEEE Transactions on Consumer Electronics* (1999)
4. Intel. *Intel Open Source Computer Vision Library*. Available at http://www.intel.com/research/mrl/research/opencv.
5. Kitahara, I. & Ohta, Y.: Scalable 3D Representation for 3D Video Display in a Large-scale Space. *Proceedings of IEEE Virtual Reality Conference* (2003)
6. Longuet-Higgins, H.: Multiple Interpretations of a Pair of Images of a Surface. *Proceedings of the Royal Society London A*, Vol. 418, Pages 1-15 (1988)
7. Luong, Q. & Faugeras, O. The Fundamental Matrix: Theory, Algorithms, and Stability Analysis. *International Journal of Computer Vision*, Vol. 17, (1995)
8. Matsuyama, T., Wu, X., Takai, T. & Wada, T.: Real-time Dynamic 3-D Object Shape Reconstruction and High-Fidelity Texture Mapping for 3-D Video. *IEEE Transactions on Circuits and Systems for Video Technology* (2004)
9. Pollefeys, M., Koch, R. & Gool, L.: A simple and efficient rectification method for general motion. *International Conference on Computer Vision* (1999)
10. Seitz, S. & Dyer, C.: View Morphing. *Proceedings of SIGGRAPH* (1996)
11. Shapiro, L. & Stockman, G.: *Computer Vision*. Prentice Hall (2001)
12. Smith, S. & Brady, J.: SUSAN - A New Approach to Low Level Image Processing. *International Journal of Computer Vision*, Vol. 23 (1997)
13. Vedulayz, S., Bakerz, S. & Kanade, T.: Spatial-temporal View Interpolation. *Proceedings of the 13th ACM Eurographics Workshop on Rendering*, (2002)
14. Zhang, L., Wang, D. & Vincent, A.: Adaptive Reconstruction of Intermediate Views from Stereoscopic Images. *IEEE Transactions on Circuits and Systems for Video Technology* (2006)

A Perceptual Framework for Comparisons of Direct Volume Rendered Images

Hon-Cheng Wong[1], Huamin Qu[2], Un-Hong Wong[1],
Zesheng Tang[1], and Klaus Mueller[3]

[1] Faculty of Information Technology,
Macau University of Science and Technology, Macao, China
[2] Department of Computer Science and Engineering,
Hong Kong University of Science and Technology, Hong Kong, China
[3] Department of Computer Science,
Stony Brook University, Stony Brook, USA
hcwong@ieee.org, huamin@cs.ust.hk, uhwong@computer.org,
ztang@must.edu.mo, mueller@cs.sunysb.edu

Abstract. Direct volume rendering (DVR) has been widely used by physicians, scientists, and engineers in many applications. There are various DVR algorithms and the images generated by these algorithms are somewhat different. Because these direct volume rendered images will be perceived by human beings, it is important to evaluate their quality based on human perception. One of the key perceptual factors is that whether and how the visible differences between two images will be observed by users. In this paper we propose a perceptual framework, which is based on the Visible Differences Predictor (VDP), for comparing the direct volume rendered images generated with different algorithms or the same algorithm with different specifications such as shading method, gradient estimation scheme, and sampling rate. Our framework consists of a volume rendering engine and a VDP component. The experimental results on some real volume data show that the visible differences between two direct volume rendered images can be measured quantitatively with our framework. Our method can help users choose suitable DVR algorithms and specifications for their applications from a perceptual perspective and steer the visualization process.

1 Introduction

Direct volume rendering (DVR) is a widely used technique in visualization, which directly renders 3D volume data into 2D images without generating any intermediate geometric primitives. There are many DVR algorithms developed in the past two decades, including ray-casting [1], splatting [2], shear-warp [3], 2D texture slicing [4], 3D texture slicing [5], and GPU-based volume rendering [4] [6] [7]. A recent survey of DVR algorithms can be found in [8].

It is well known that direct volume rendered images generated by different DVR methods are somewhat different and some algorithms can generate images with better quality than others. Therefore, there is a need to compare the quality of direct volume rendered images generated by different methods and specifications. Fortunately,

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1314–1323, 2006.

there are more and more works reporting the comparisons of volume rendering algorithms [9] [10] and volume rendered images [11].

These direct volume rendered images will be perceived by the human beings. Therefore, it is important to quantitatively evaluate DVR images based on human perception. One of the key perceptual factors is that whether the visible differences between two images will be observed. However, research concerning this factor is scant. In this paper we propose a perceptual framework, which is based on Daly's Visible Differences Predictor [12], for comparative study of direct volume rendered images. It will be used to predict the visible differences of the direct volume rendered images generated with different algorithms and the same algorithm with different specifications.

The remainder of this paper is organized as follows: We first introduce related work in Section 2. We then describe our framework and review VDP in Section 3. Next, we compare the direct volume rendered images by using our framework in Section 4. Finally, we conclude our work and discuss future research directions in Section 5.

2 Related Work

Comparative evaluation in DVR algorithms: Methods for comparing DVR algorithms can be categorized into three classes:
1) Image level methods [13] [9]: They usually compare DVR images side-by-side using various methods such as difference image, mean square errors (MSE), and root mean square error (RMSE); 2) Data level methods [10]: They use raw data and intermediate information obtained during the rendering process for comparison; 3) Analytical methods to calculate the error bounds [14]: They analyze the errors in gradient calculations, normal estimation schemes, and filtering and reconstruction operations. In addition to these comparative methods, Mei"sner et al. [15] performed a practical evaluation of popular DVR algorithms in terms of rendering performance on real-life data sets.
Perception issues in computer graphics: Considerable concern has arisen over the perception in graphics research in recent years, especially in the area of global illumination. There has been much work on perceptually-based rendering proposed. Most of them focus on two tasks: 1) To establish stopping criteria for high quality rendering systems by developing perceptual metrics [16] and 2) To optimally manage resource allocation for efficient rendering algorithms by using perceptual metrics [17] [18]. In addition, Rushmeier et al. [19] proposed some metrics for comparing real and synthetic images.
Perception issues in visualization: There is a growing number of research on using perception for visualization. For example, Lu et al. [20] utilized several feature enhancement techniques to create effective and interactive visualizations of scientific and medical data sets. Ebert [21], Interrante [22], and Chalmers and Cater [23] recently give excellent surveys on perception issues in visualization. Zhou et al. [24] presented a study of image comparison metrics for quantifying the magnitude of difference between a visualization of a computer simulation and a photographic image captured from an experiment.

To the best of our knowledge, there has been little work which applies perception in comparative evaluation of DVR. Inspired by the work of Myszkowski [18] which uses

VDP for global illumination problems, we have developed a framework based on VDP to compare the direct volume rendered images.

3 Our Framework and Visible Differences Predictor (VDP)

The block diagram of our framework is shown in Figure 1. Our framework contains a volume rendering engine which generates the images by using different DVR algorithms supported in the engine. Two direct volume rendered images are produced with the user specified settings. Then they will be sent to the VDP, which compares these images and gives a differences map (VDP responses) as output.

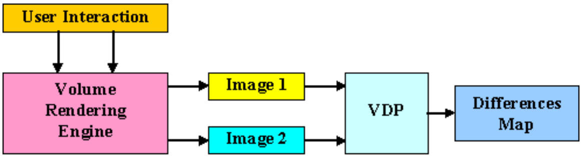


Fig. 1. Block diagram of our framework

There are many metrics based on Human Visual System (HVS). Two most popular ones are Daly's Visible Differences Predictor (VDP) [12] and Sarnoff Visual Discrimination Model (VDM) [25]. Both metrics were shown to perform equally well on average [26]. We chose VDP in our framework because of its modularity and extensibility. The block diagram of VDP [18] is shown in Figure 2. VDP receives as input a pair of images (target image and mask image), and outputs a map of probability values, which indicates how the differences between those images are perceived [12] [18]. These two input images are first processed by the *amplitude nonlinearity*, which simulates the adaptation of HVS to local luminance. Then the resulting images are converted into frequency domain using FFT. After that, *contrast sensitivity function (CSF)*, which simulates the variations in visual sensitivity of HVS, is performed on the frequency signals. These images are then converted to spatial frequency and orientation channels using a pyramid-style *cortex transform*. *Masking function* which is used to increase the

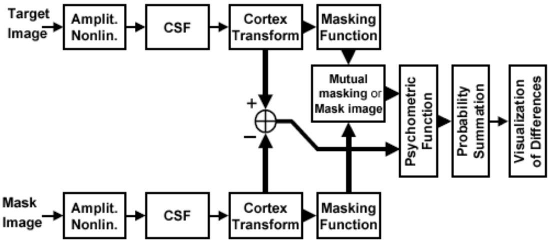


Fig. 2. Block diagram of the Visible Differences Predictor [18]

threshold of detectability is applied to these images and the minimal threshold elevation value for corresponding channels and pixels are taken by *mutual masking/mask image*. A *psychometric function* for predicting the probability of perceived differences is applied to these images and finally the predicted probability is visualized.

As VDP is a general purpose predictor of the differences between images, it can be used to evaluate pairs of images for a wide range of applications. Although VDP does not support chromatic channels in input images, in DVR applications many important insights such as depth cues can be well captured in an achromatic images. Thus we embedded VDP in our framework for comparing the direct volume rendered images. Since the HVS is more sensitive to the differences of contrast and less sensitive to the differences between colors, we convert color direct volume rendered images generated by the volume rendering engine to gray-scale images before sending them to the VDP.

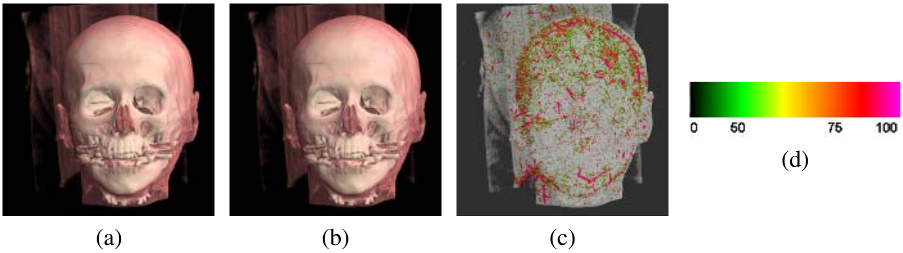
All comparisons were performed on a 2.4GHz AMD Opteron 280 processor with 6GB main memory, and an NVIDIA Quadro FX 4500 graphics card with 512MB video memory. The resulting VDP responses (differences map in our framework) are represented as a color map which is blended with the original grey-scale target image. Color is added to each pixel in this target image to indicate its difference detection probability values. The probability values greater than 0.75, which is the standard threshold value for discrimination tasks [27], are set to red pixels. In the rest of the paper, we usually provide results in a set of three figures. The first two figures are generated using different algorithms or settings and the third one shows their differences map, which is encoded in the same color scale as in Figure 3 (d). The background pixels (black pixels) are not included in the calculation of percentage of red pixels in the VDP result.

4 Comparisons of Direct Volume Rendered Images

In this section, we use our framework to measure the perceptible differences in the direct volume rendered images generated with different algorithms or the same algorithm with different specifications. For DVR algorithms we select two most popular methods: GPU-based ray-casting [6] and 3D texture slicing [5]. And several specifications including shading, gradient estimation scheme, and sampling rate are chosen for comparisons. We limit our case studies to static, regular or rectilinear, scalar volume data only. The size of all images is 512×512. All algorithm-independent parameters such as viewing, transfer functions, and optical model, are kept constant in each image comparison set in order to have a fair comparison. Experimental results will be discussed at the end of this section.

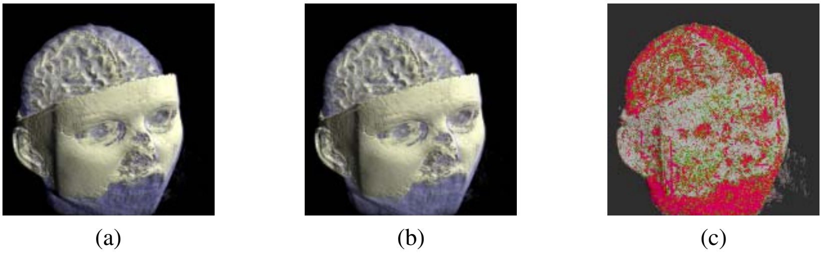
4.1 GPU-Based Ray-Casting Versus 3D Texture Slicing

Two data sets are used here: 256^3 CT human head and 256^3 MRI human head. Figure 3 and Figure 4 compare the direct volume rendered images generated with GPU-based ray-casting and 3D texture slicing for these two data sets respectively. From the VDP responses shown in Figure 3 (c), we can see that there are some noticeable areas of red pixels, which indicate the differences between Figure 3 (a) and (b) in these regions are quite noticeable (probability $> 75\%$) to human beings. And there are only some light green pixels inside the regions of red pixels. The percentage of red pixels is 11.89% in the VDP result. In Figure 4 (c), we can see some pieces of red pixels and many small pieces of light greed pixels on the surface of the MRI head. The percentage of red pixels is 28.27% in it. In both Figure 3 (c) and Figure 4 (c), the red pixels are distributed in the transparent regions. The VDP predictions (Figure 3 (c) and Figure 4 (c)) coincide with the human perception of the visual results shown in Figure 3 (a) and (b), and Figure 4 (a) and (b).



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

Fig. 3. Comparison of GPU-based ray-casting and 3D texture slicing (256^3 CT human head): (a) GPU-based ray-casting; (b) 3D texture slicing; (c) VDP result (Red pixels: 11.89%); (d) Color scales for encoding the probabilities (%) in (c)



(a) (b) (c)

Fig. 4. Comparison of GPU-based ray-casting and 3D texture slicing (256^3 MRI human head): (a) GPU-based ray-casting; (b) 3D texture slicing; (c) VDP result (Red pixels: 28.27%)

4.2 Shading

We are interested in the visual differences between the rendering results in the following two scenarios. First, in pre-shaded DVR, the shading model at the grid samples is evaluated first and then the illumination is interpolated. In contrast, the normal is interpolated first and then the shading model for each reconstructed sample is evaluated in post-shaded DVR. Second, separate color interpolation was used in [1], which interpolates voxel colors and opacities separately before computing the product of them. Wittenbrink et al. [28] pointed out that it is more correct to multiply color and opacity beforehand at each voxel and then interpolate the product. We compare the pre-shaded and post-shaded DVR images, as well as the opacity-weighted color interpolated [28] and separate color interpolated DVR images using the GPU-based ray-caster with a 256^3 engine data set.

Figure 5 shows the comparison of pre-shaded and post-shaded DVR. From Figure 5 (c) we can know that the rendering results between pre-shaded (Figure 5 (a)) and post-shaded (Figure 5 (b)) DVR images are quite different. The percentage of red pixels reaches 54.79%, which means that such differences in these two images are very noticeable to human beings.

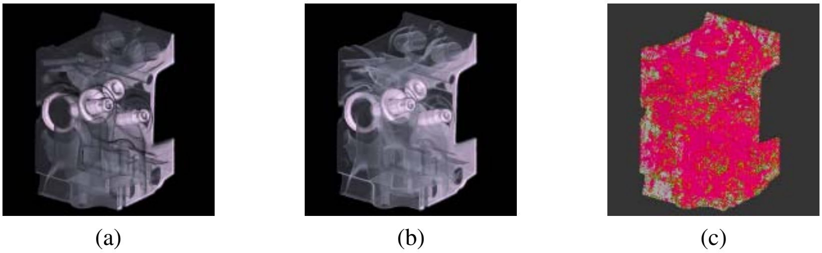


Fig. 5. Comparison of pre-shaded DVR and post-shaded DVR (256^3 engine): (a) Pre-shaded DVR; (b) Post-shaded DVR; (c) VDP result (Red pixels: 54.79%)

Figure 6 shows the comparison of opacity-weighted color interpolated and separate color interpolated DVRs. The percentage of red pixels is 64.00%. And the distribution of the most noticeable differences between Figure 6 (a) and (b) can be easily found in Figure 6 (c).

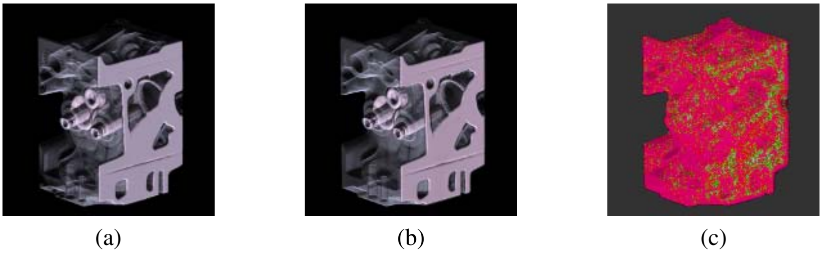


Fig. 6. Comparison of opacity-weighted color interpolated and separate color interpolated DVRs (256^3 engine): (a) Opacity-weighted color interpolated DVR; (b) Separate color interpolated DVR; (c) VDP result (Red pixels: 64.00%)

4.3 Gradient Estimation Scheme

In DVR, different gradient estimation scheme expresses the choice of normal computation from the volume data. They may significantly affect the shading and appearance of the rendering results. Two schemes are compared in Figure 7: central difference operator [1] which computes gradients at data values in the *x*, *y*, *z* direction and then uses the gradients at the eight nearest surrounding data locations to interpolate the gradient vectors for locations other than at data locations; intermediate difference operator [29] which computes gradient vectors situated between data locations using differences in data values at the immediate neighbors. Figure 7 (c) shows that the differences between the images are very obvious. The percentage of red pixels is 50.62%.

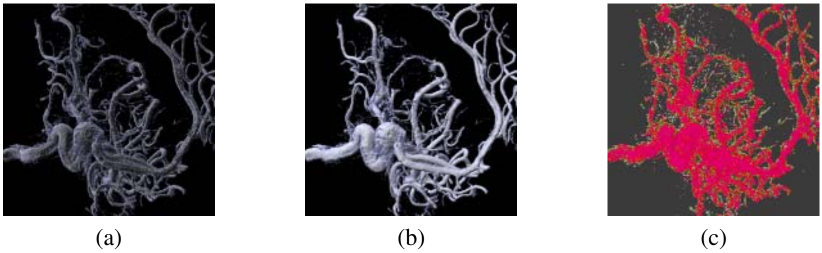


Fig. 7. Comparison of intermediate and central difference operators in DVR (256^3 aneurism): (a) With intermediate difference operator; (b) With central difference operators; (c) VDP result (Red pixels: 50.62%)

4.4 Sampling Rate

Direct volume rendered images generated by a GPU-based ray-caster with different sampling rates are compared in this section. The motivation for performing this kind of comparisons is that we want to reduce the rendering time of the data sets by downgrading their sampling rates without sacrificing the image quality too much. A 256^3 CT head data set is used and two sets of comparison are shown in Figure 8, where the upper set compares images generated with 512 samples and with 640 samples, and the lower set compares images generated with 1280 samples and 1408 samples. Figure 8 (c) shows that the differences between Figure 8 (a) and (b) are quite noticeable, and the percentage of red pixels is 39.54%. From Figure 8 (f), we can find that the differences between Figure 8 (d) and (e) are not so noticeable, where the percentage of red pixels is 28.46%.

4.5 Discussions

The experimental results show that differences between two direct volume rendered images are quite noticeable in transparent regions, indicating that different DVR algorithms or the same algorithm with different specifications are sensitive to these regions because of the inner structures of the data visualized there. Thus the choice of DVR algorithms or specifications have major impact on the visual result of the transparent regions. For shading methods, the visual appearance of images generated with pre-shaded and post-shaded are quite different as shown in the VDP result (Figure5 (c)). The image quality of separate color interpolation is considered having color-bleeding artifacts [28]. The VDP result provides a distribution of such artifacts that may be noticed in the regions with red pixels. For gradient estimation schemes, the intermediate difference operator in DVR offers a better shading of the images [29] and the VDP result indicates such differences clearly. For sampling rates, the differences between two images with higher sampling rates are less than those two images with lower sampling rates. With enough high sampling rates, further increasing the sampling rate may not improve the image quality. With our framework, the differences between direct volume rendered images can be easily identified quantitatively. Thus it may be used for researchers to determine an appropriate DVR algorithm or a set of specifications for their research and applications.

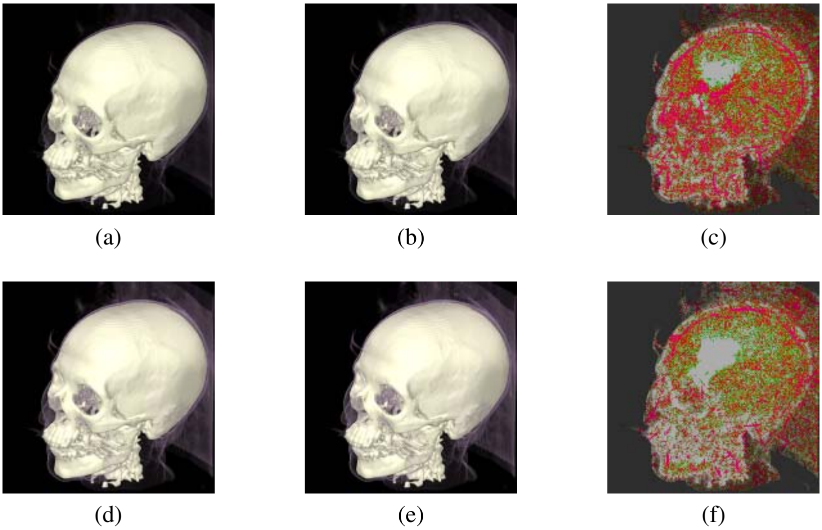


Fig. 8. Comparison of direct volume rendered images with different sampling rates (256^3 CT head): (a) Image rendered with 512 samples; (b) Image rendered with 640 samples; (c) VDP result of (a) and (b) (Red pixels: 39.54%); (d) Image rendered with 1280 samples; (e) Image rendered with 1408 samples; (f) VDP result of (d) and (e) (Red pixels: 28.46%)

5 Conclusions and Future Work

In this paper, we proposed a framework for comparing direct volume rendered images generated by different algorithms or the same algorithm with different specifications. Two most popular DVR algorithms, GPU-based ray-casting and 3D texture slicing, are selected for comparisons. Some specifications including shading methods (pre-shaded *v.s.* post-shaded, separate color interpolated *v.s.* opacity-weighted color interpolated), gradient estimation schemes (central and intermediate difference operators), and sampling rate are also compared. The experimental results with real data sets show that we can get quantitative and perceptual comparison results with our framework. To conclude, this study is our first attempt to apply perception knowledge on direct volume rendered images. It will allow scientists and engineers to better understand volume data.

In the future, we would like to perform a psychophysical validation of VDP for DVR applications and use our framework to conduct a more comprehensive study involving more direct volume rendered images generated by different kernels such as different filters and optical models. As the computation of VDP is quite expensive due to the multiscale spatial processing involved in some of its components, we plan to implement VDP on GPUs and integrate it with existing GPU-based volume rendering algorithms into our framework to provide fast feedbacks of comparison results. In addition, VDP may be used for level-of-detail (LOD) selection in large volume visualization as what Wang et al. [30] have done recently. This fast comparative framework can then be used for evaluating direct volume rendered images from a perceptual point of view and steering the DVR process.

Acknowledgments. This work is supported by the Science and Technology Development Fund of Macao SAR grant 001/2005/A.

References

1. Levoy, M.: Display of surfaces from volume data. IEEE Computer Graphics and Applications **8** (1988) 29–37
2. Westover, L.: Footprint evaluation for volume rendering. In: Computer Graphics (ACM SIGGRAPH '90). Volume 24. (1990) 367–376
3. Lacroute, P., Levoy, M.: Fast volume rendering using a shear-warp factorization of the viewing transformation. In: Proceedings of ACM SIGGRAPH '94. (1994) 451–458
4. Rezk-Salama, C., Engel, K., Bauer, M., Greiner, G., Ertl, T.: Interactive volume rendering on standard pc graphics hardware using multi-textures and multi-stage rasterization. In: Proceedings of ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware '00. (2000) 109–118
5. Gelder, A.V., Kim, K.: Direct volume rendering with shading via 3d textures. In: Proceedings of ACM/IEEE Symposium on Volume Visualization '96. (1996) 22–30
6. Krüger, J., Westermann, R.: Acceleration techniques for gpu-based volume rendering. In: Proceedings of IEEE Visualization '03. (2003) 287–292
7. Xue, D., Crawfis, R.: Efficient splatting using modern graphics hardware. Journal of Graphics Tools **8** (2003) 1–21
8. Kaufman, A., Mueller, K.: Overview of volume rendering. In: The Visualization Handbook, Edited by C. D. Hansen and C. R. Johnson, Elsevier Butterworth-Heinemann (2005) 127–174
9. Williams, P.L., Uselton, S.P.: Metrics and generation specifications for comparing volume-rendered images. Journal of Visualization and Computer Animation **10** (1999) 159–178
10. Kim, K., Wittenbrink, C.M., Pang, A.: Extended specifications and test data sets for data level comparisions of direct volume rendering algorithms. IEEE Transactions on Visualization and Computer Graphics **7** (2001) 299–317
11. Pommert, A., Höhne, K.H.: Evaluation of image quality in medical volume visualization: The state of the art. In: Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention '02. (2002) 598–605
12. Daly, S.: The visible differences predictor: An algorithm for the assessment of image fidelity. In: Digital Image and Human Vision, Edited by A. B. Watson, The MIT Press (1993) 179–206
13. Gaddipatti, A., Machiraju, R., Yagel, R.: Steering image generation with wavelet based perceptual metric. Graphics Graphics Forum (EUROGRAPHICS '97) **16** (1997) 241–251
14. Machiraju, R., Yagel, R.: Reconstruction error characterization and control: A sampling theory approach. IEEE Transactions on Visualization and Computer Graphics **2** (1996) 364–378
15. Mei"sner, M., Huang, J., Bartz, D., Mueller, K., Crawfis, R.: A practical evaluation of popular volume rendering algorithms. In: Proceedings of IEEE Symposium on Volume Visualization '00. (2000) 81–90
16. Ramasubramanian, M., Pattanaik, S.N., Greenberg, D.P.: A perceptually based physical error metric for realistic image synthesis. In: Proceedings of ACM SIGGRAPH '99. (1999) 73–82
17. Bolin, M., Meyer, G.: A perceptually based adaptive sampling algorithm. In: Proceedings of ACM SIGGRAPH '98. (1998) 299–309
18. Myszkowski, K.: The visible differences predictor: Applications to global illumination problems. In: Proceedings of EUROGRAPHICS Workshop on Rendering '98. (1998) 223–236

19. Rushmeier, H., Ward, G., Piatko, C., Sanders, P., Rust, B.: Comparing real and synthetic images: Some ideas about metrics? In: Proceedings of EUROGRAPHICS Workshop on Rendering '95. (1995) 82–91
20. Lu, A., Morris, C.J., Taylor, J., Ebert, D.S., Hansen, C., Rheingans, P., Hartner, M.: Illustrative interactive stipple rendering. IEEE Transactions on Visualization and Computer Graphics **9** (2003) 127–138
21. Ebert, D.S.: Extending visualization to perceptualization: The importance of perception in effective communication of information. In: The Visualization Handbook, Edited by C. D. Hansen and C. R. Johnson, Elsevier Butterworth-Heinemann (2005) 771–780
22. Interrante, V.: Art and science in visualization. In: The Visualization Handbook, Edited by C. D. Hansen and C. R. Johnson, Elsevier Butterworth-Heinemann (2005) 781–806
23. Chalmers, A., Cater, K.: Exploiting human visual perception in visualization. In: The Visualization Handbook, Edited by C. D. Hansen and C. R. Johnson, Elsevier Butterworth-Heinemann (2005) 807–816
24. Zhou, H., Chen, M., Webster, M.F.: Comparative evaluation of visualization and experimental results using image comparison metrics. In: Proceedings of IEEE Visualization '02. (2002) 315–322
25. Lubin, J.: A visual discrimination model for imaging system design and evaluation. In: Vision Models for Target Detection and Recognition, Edited by E. Peli, World Scientific Publishing (1995) 245–283
26. Li, B., Meyer, G., Klassen, R.: A comparison of two image quality models. In: Human Vision and Electronic Imaging III, SPIE Vol. 3299 (1998) 98–109
27. Wilson, H.R.: Psychophysical models of spatial vision and hyperacuity. In: Spatial Vision, Vol. 10, Vision and Visual Disfunction, Edited by D. Regan, MIT Press (1991) 64–86
28. Wittenbrink, C.M., Malzbender, T., Goss, M.E.: Opacity-weighted color interpolation for volume sampling. In: Proceedings of Symposium on Volume Visualization '98. (1998) 135–142
29. van Scheltinga, J.T., Bosma, M., Smit, J., Lobregt, S.: Image quality improvements in volume rendering. In: Proceedings of the Fourth International Conference on Visualization in Biomedical Computing (VBC '96). (1996) 87–92
30. Wang, C., Garcia, A., Shen, H.W.: Interactive level-of-detail selection using image-based quality metric for large volume visualization. IEEE Transactions on Visualization and Computer Graphics (accepted) (2006)

Hierarchical Contribution Culling for Fast Rendering of Complex Scenes

Jong-Seung Park and Bum-Jong Lee

Department of Computer Science & Engineering, University of Incheon,
177 Dohwa-dong, Nam-gu, Incheon, 402-749, Republic of Korea
{jong, leeyanga}@incheon.ac.kr

Abstract. An intelligent culling technique is necessary in rendering 3D scenes to allow the real-time navigation in complex virtual environments. When rendering a large scene, a huge number of objects can reside in a viewing frustum. Virtual urban environments present challenges to interactive visualization systems, because of the huge complexity. This paper describes a LOD management (or contribution culling) method for the view-dependent real-time rendering of complex huge urban scenes. From the experimental results, we found the proposed LOD management method effectively limits the amount of objects to put into the rendering pipeline without notable degradation of rendering quality.

1 Introduction

A huge amount of previous works have been focused on interactive visualization of complex virtual environments [1][2]. A large number of objects should be drawn to ensure the high quality rendering. However, it slows down the rendering speed, which is not acceptable for many interactive applications. Some insignificant objects need not be drawn and they should be culled away from the set of objects in the scene. Such a culling process should be proceeded fast enough to satisfy the real-time requirement.

A suitable spatial data structure is required for the fast culling. The spatial data structures can be classified into four categories [3][4]: *Bounding volume hierarchies*, *Binary space partitioning (BSP) trees*, *Octrees*, and *Quadtrees*. A bounding volume is a volume that encloses a set of objects. Bounding volumes constitute a hierarchical structure where a bounding volume in a high-level node corresponds to the bounding volume for a large cluster of objects and that in a lowest-level node corresponds to the bounding volume for a single object. At the rendering, a large number of objects can be culled out at once using the hierarchical structure of bounding volumes. A binary space partitioning (BSP) tree is a data structure that represents a recursive hierarchical subdivision of a 3D space into convex subspaces. The BSP tree partitions a space by any hyperplane that intersects the interior of that space. The result is two new subspaces that can be further partitioned recursively. An octree is a data structure which can define a shape in 3D space where the 3D space is divided into eight cubes. It can

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1324–1333, 2006.

be further divided into eight smaller cubes recursively. This can be represented by a tree structure where each node corresponds to a cube in the space.

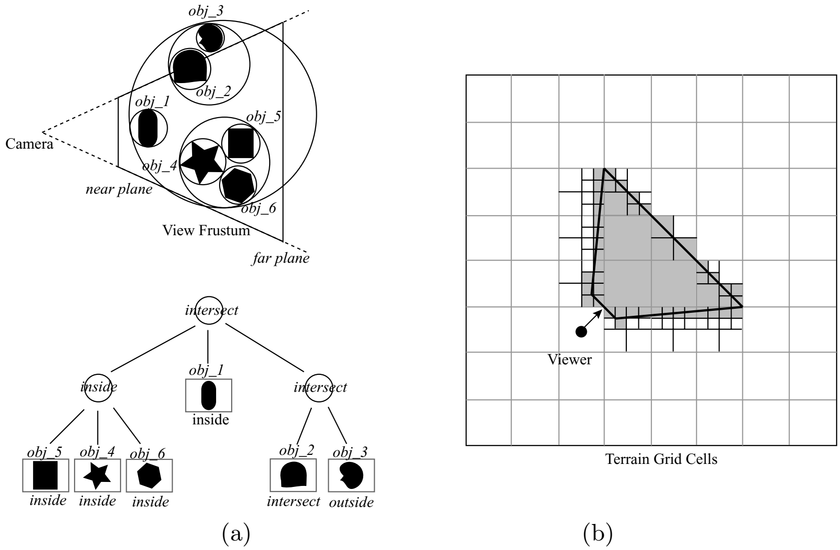
The quadtree is a tree structure in which each internal node has up to four children. Quadtrees are most often used to partition a 2D space by recursively subdividing it into four quadrants or regions. The regions may be square or rectangular. Quadtree traversal is required for the visibility test [5]. To reduce the traversal complexity, several efficient quadtree traversal techniques have been proposed such as a block-priority-based traversal [6]. Quadtrees could also be used to improve rendering quality [7].

Geometry culling is concerned with removing objects that a viewer cannot see in navigation. It reduces the number of polygons sent down the rendering pipeline by excluding faces that need not be drawn. There are several different kinds and stages of culling [3]. The view frustum culling is a form of visibility culling based on the premise that only objects in the current view volume must be drawn [8]. Objects outside the volume will not be visible to the viewer, so they should be discarded. The backface culling can be considered as a part of the hidden surface removal process of a rendering pipeline. In the backface culling, polygons (usually triangles) that are not facing the camera are removed from the rendering pipeline. This is simply done by comparing the polygon's surface normal with the position of the camera. The portal culling divides the models into cells and portals, computing a conservative estimate of which cells are visible at the render time. The occlusion culling attempts to determine which portions of objects are hidden by other objects from a given viewpoint. The occluded objects are discarded before they are sent to the rendering pipeline. The occlusion culling also are a form of visible surface determination algorithms that attempt to resolve visible (or non-visible surfaces) at larger granularity than pixel-by-pixel testing [9].

In this paper, we describes a simple and fast contribution culling technique using a hierarchical tree structure. If the polygon is too far from the viewer it will not be drawn since its contribution to the rendering quality is not significant. Hence, we throw away all objects whose projected area into the screen space is below the threshold. In Section 2, we describe the hierarchical tree structure for the LOD management. In Section 3, we explain the fast contribution culling and rendering method and provides experimental results using our complex virtual urban environments. Finally, concluding remarks are described in Section 4.

2 Hierarchical Representation of Scene Structure

For the real-time rendering for complex scenes, the use of *levels-of-detail* (LODs) is necessary. The LOD management involves decreasing the details of an object as it moves away from the viewer. The use of LOD increases the efficiency of rendering by reducing the number of polygons to be drawn. Real-time rendering issues have been actively studied to satisfy the requirement of real-time computation in drawing large complex scenes. A distributed visibility culling technique was proposed to render large complex scenes using occluders with the BSP tree[10].



(a) (b)

Fig. 1. (a) A view frustum with objects and the corresponding bounding volume hierarchy, (b) Subdivision of terrain grid cells

Bittner divided a space using space subdivision of a line space, and rendered visible objects by selecting an occluder[11]. Cohen-or[12] described a method of partitioning the view space into cells containing a conservative superset of the visible objects.

The view frustum culling is the easiest culling. If a polygon is not in the field of view then it need not be rendered. In the culling, it is required to test the intersection of the polygon and the six planes of the view frustum. The temporal coherence and the camera movement coherence can be exploited to reduce the intersection tests. The temporal coherence is based on the fact that if a bounding volume of a node was outside one of the view frustum plane at the previous frame, then the node is probably outside the same plane. The camera translation and rotation coherency exploits the fact that, when navigating in a 3D space, only rotation around an axis or translation is commonly applied to the camera. In those cases, many objects that were outside the plane at the last frame still remain outside.

Fig. 1(a) shows a view frustum hierarchy. If a non-leaf node is inside the view frustum, all the child nodes of it are also inside the view frustum. This hierarchical structure speeds up the culling process. An appropriate representation for the effective view frustum culling implementation is a quadtree [13][14][15]. The terrain cell partitioning in the quadtree-based view frustum culling is shown in Fig. 1(b). The terrain area is divided into cells of regular squares. A bounding box centered at the camera position is managed during the camera movement. Only when the camera crosses the bounding box, the system recomputes the quadtree structure. When the cells are on the frustum boundary, low level quadtree nodes

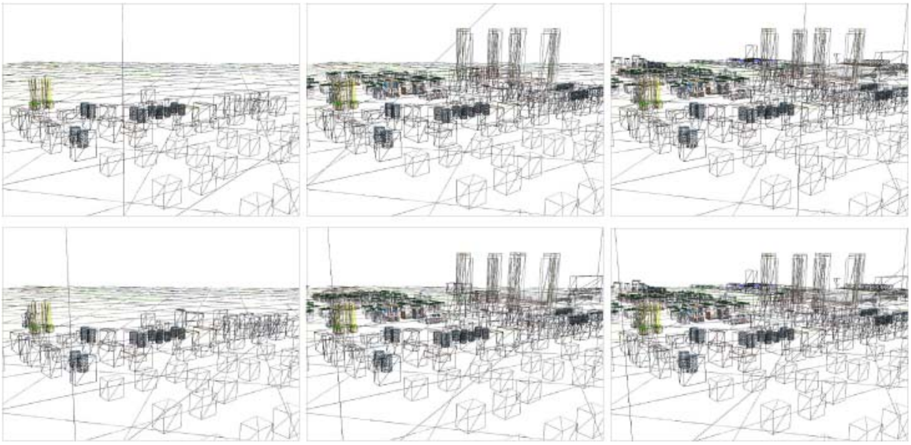


Fig. 2. Wire-frame views of the rendered scenes using a simple culling by the number of polygons (upper) and using a simple culling by the distance measure (lower)

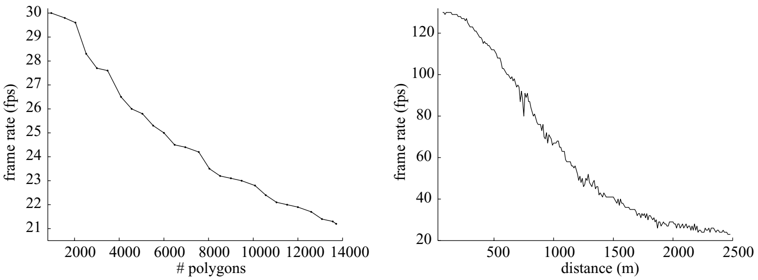


Fig. 3. Rendering frame rate for the simple methods according to the number of polygons (left) and the visibility range (right)

are used to refine the areas in order to reduce the number of objects to be sent to the rendering pipeline.

Several techniques using hierarchical data structures have been proposed to efficiently handle a large amount of data. Tsuji [16] proposed a method of combining the dynamic control of LOD and the conservative occlusion culling by using a new hierarchical data structure. Face clusters are recursively partitioned to dynamically control the LOD.

We use the hierarchical weighted tree structure for the efficient implementation of the view frustum culling and the contribution culling. When rendering a complex environment containing a huge number of objects, the rendering speed should not be slowed down. To avoid such dependency on the number of visible objects, we investigated a criteria to choose the most contributive objects for the high quality rendering. Small objects in the screen space are culled out even though they are near to the camera position.

3 View Space Partitioning and Contribution Culling

Our goal is to guarantee high-fidelity rendering of large models using a low computational load. Hence, it is necessary to reduce the number of polygons sent to the rendering pipeline. We select only the fixed number of the most important visible objects and send them to the rendering pipeline. If small details contribute little or nothing to the scene fidelity then it can be discarded.

The contribution of each object is measured as the area of the projection in the screen space. For each object its bounding volume is computed at the initial loading stage. When creating or updating the hierarchical structure, the bounding volume is projected onto the projection plane and the area of the projection is then estimated in pixels. The area is a measure of the contribution of the object. If the number of pixels in the area is smaller than a given threshold value then the object is culled away from the rendering pipeline.

Our view space partitioning and contribution culling algorithm is as follows. As an initialization stage, we find the quadrilateral visible region of the terrain in the image space. The region corresponds to the root node of the hierarchical structure. Initially, the weight of the node is set to zero. Beginning from the node, we recursively split the region into four subregions. For each node the four corners of the region are always kept. Since we knows the view frustum geometry and the projection matrix, we can find the world coordinates of the corners. If the number of objects in a region is equal to zero or one, we do not split the region any more. If there are multiple objects in the region we split the region into four subregions. For each subregion we also create a node and attach it to the parent node.

For each newly created node we evaluate its weight and associate the weight to the node. The weight of a node is defined as the sum of all weights for objects in inside the screen rectangular region corresponding to the node. Note that the weight of an object is the number of pixels of the bounding rectangle in the image space. When a node is created and its weight is evaluate, the weights of all its parent nodes are also updated by adding the weight of the new node.

The hierarchical structure is updated only when there is a significant change in camera position or orientation. In a rendering cycle, only a fixed number of significant objects are rendered. The number of objects to be rendered depends on the desired FPS for a specific machine. To select the significant objects we traverse the tree in a breath-first-order.

Let N be the desired number of the objects to be drawn. Then, we try to choose the most N significant objects starting from the root node n_1. If the node have subnodes, we try to choose the N significant objects from the subnodes. For a subnode n_k we try to choose $\lfloor(M * w_k/w_1 + 0.5)\rfloor$ objects where w_k is the weight of the node n_k. Note that the sum of all subnode weights is equal to the weight of the parent node. The selection process is repeated recursively until the desired number of objects are chosen or there are no more significant objects in the subnodes.

We have tested the proposed method on a complex virtual urban environment. Our rendering engine is based on the MS Windows XP platform and is

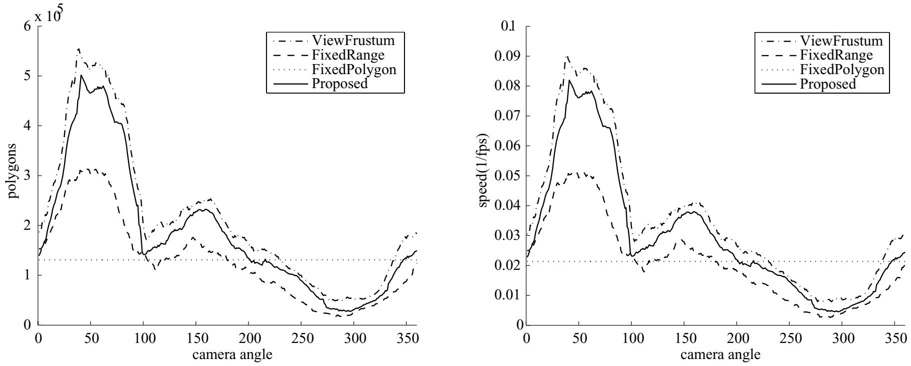


Fig. 4. Comparison of rendering performance for the 360° rotating camera

implemented using C++ and DirectX 9.0. The machine has a 2.8GHz Pentium 4 processor with 512MB DDR RAM and an ATI RADEON 9200 GPU with 64MB DRAM.

We implemented four different methods and compared the fidelity of the rendering and the frame rate. The methods are denotated by *ViewFrustum*, *FixedRange*, *FixedPolygon*, and *Proposed*, each corresponds to the view frustum culling method, the fixed visibility range method, the fixed number of polygons method, and the proposed method, respectively. The method *ViewFrustum* renders only the objects that are inside the view frustum. The method *FixedRange* culls away objects that are placed outside the distance over a specified distance. The method *FixedPolygon* limits the number of polygons to be sent to the rendering pipeline. Only the given number of the nearest polygons to the viewer are sent to the pipeline. The method *Proposed* chooses the given number of the most significant objects and send them to the pipeline.

We tested the proposed method by applying it to render a huge number of structures in the Songdo area of the Incheon Free Economic Zone, a metropolitan section which is currently under development. It will occupy a vast stretch of land totaling 13,000 acres by the time all development is completed. Based on the construction implementation plan, we modeled the virtual Songdo city. Currently, our virtual environment contains over a million objects for high rise landmark business buildings, hotels, research centers, industrial complex, bio complex, residential complex buildings, and many other kinds of structures.

We first tested the two simplest simplification methods, *FixedPolygon* and *FixedRange*. Their wire-frame views of the rendered scenes are shown in Fig. 2. In the upper figure, the corresponding number of the limited polygons are 960, 3,000, and 5,040, respectively, from left to right order. In the lower figure, the corresponding visibility ranges are 300, 1,000, and 2,750 meters, respectively, from left to right order. The frame rate is shown in Fig. 3 when controlling the number of polygons (left) and the visibility range (right).

The rendering frame rate should be stable when the viewer is moving freely in a world space. We measured the number of polygons and the rendering frame

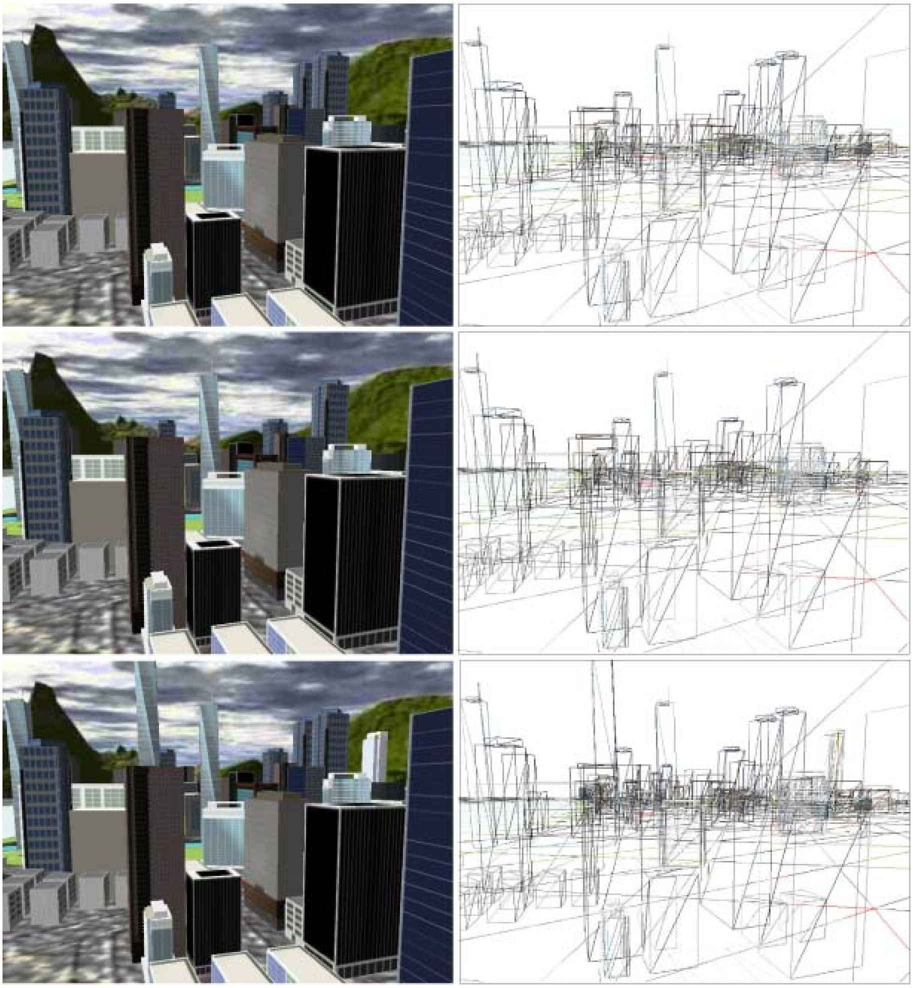


Fig. 5. Comparison of rendering fidelity: *FixedRange*, *FixedPolygon*, and *Proposed*

rate when the camera is rotated with respect to Y-axis in a fixed position. Fig. 4 shows the number of polygons (left) and the frame rate (right), respectively, for the corresponding camera angle. The proposed method has provided almost the same quality rendering results with almost the same rendering frame rate, which indicates that the computation load is not dependent on the complexity of the current view.

We compared the rendering fidelity for three methods: *FixedRange*, *FixedPolygon*, and *Proposed*. Two rendered scenes are shown in Fig. 5 and Fig. 6. The three rows in each figure correspond to *FixedRange*, *FixedPolygon*, and *Proposed*, respectively, in order from top to bottom. In the *FixedRange* method, the objects that are placed outside the distance over 900 meters do not be rendered. In the *FixedRange* method, we limit the number of polygons to 82,720 polygons. The

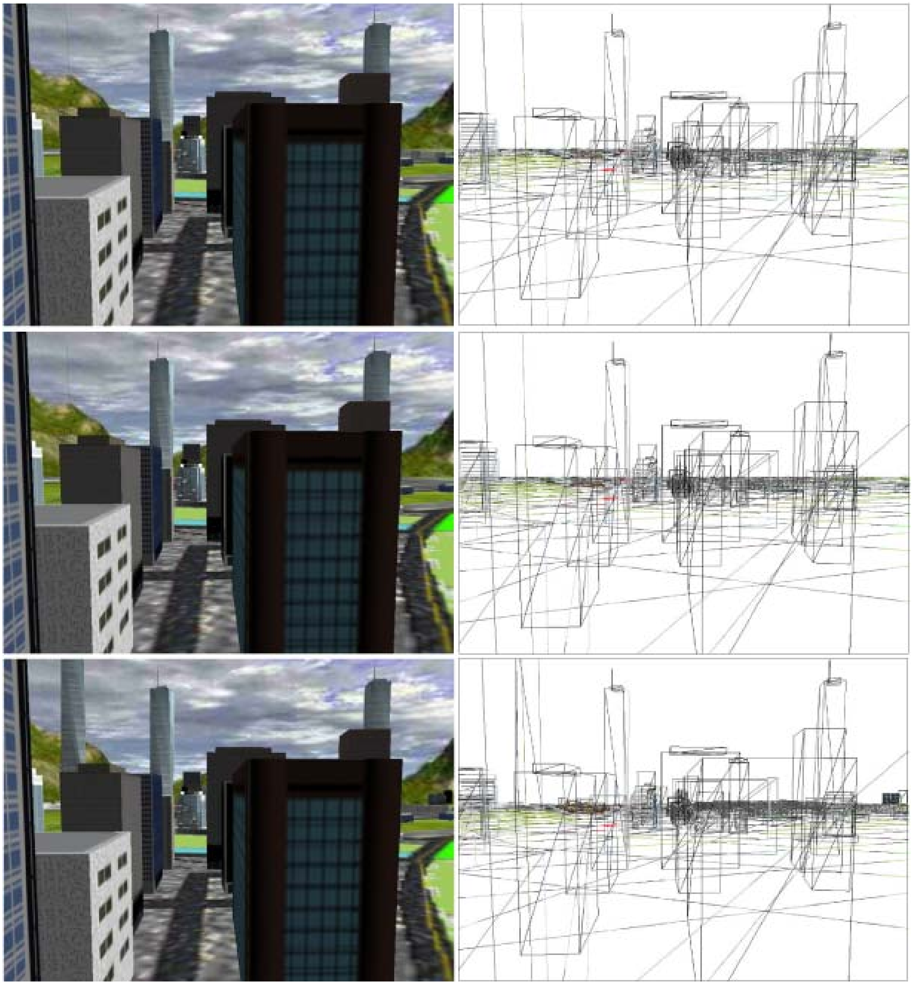


Fig. 6. Comparison of rendering fidelity: *FixedRange*, *FixedPolygon*, and *Proposed*

rendered scenes from the *Proposed* method (the third row) show that some significant buildings were rendered, which were not rendered in the other methods. In our methods, huge number of tiny objects or mostly occluded objects are not drawn. Instead, significant objects in far distance are drawn.

The proposed method has kept high-quality rendering without significant performance depreciation. Table 1 and Table 2 show the comparison of the number of polygons and the frame rate for the four different methods. In the *FixedNumber* method, the number of polygons is fixed to 82,720. In the *Proposed* method, the number of polygons is always equal to or less than that of the *FixedNumber* method. However, the rendering quality is as good as that of not using the

Table 1. Comparison of the number of polygons

methods	*View1*	*View2*	*View3*	*View4*	*View5*	*View6*
ViewFrustum	106,032	382,016	187,248	317,344	560,992	88,736
FixedRange	84,936	82,720	186,496	177,472	287,264	75,952
FixedNumber	82,720	82,720	82,720	82,720	82,720	82,720
Proposed	78,654	82,316	82,720	82,720	82,720	62,414

Table 2. Comparison of frame rates (unit: frames per second)

methods	*View1*	*View2*	*View3*	*View4*	*View5*	*View6*
ViewFrustum	58	37	51	40	21	61
FixedRange	63	62	51	52	42	64
FixedNumber	62	62	62	62	62	62
Proposed	64	61	61	61	61	64

ViewFrustum method. In our methods, huge number of tiny objects or mostly occluded objects are not drawn. Instead, we draw other significant objects.

4 Conclusion

2We have presented a fast rendering method for complex urban scenes. The method combines visibility culling and model simplification into a single framework. Using a novel contribution culling technique less significant objects are culled away from the rendering pipeline. Since only a fixed number of objects are tried to be sent to the pipeline, computation load does not depend on the scene complexity. The perceived quality of the proposed rendering method is almost same as view frustum culling that draws all visible objects.

We assume that, if the screen projected area of an object is small, then its contribution to the human perception is also small. Contributions of objects are represented by weights on nodes of the hierarchical tree structure. By traversing nodes, a fixed number of significant visible nodes are selected and sent to the pipeline. The rendering speed is determined by the number of significant objects to be drawn and, hence, we obtained a uniform frame rate in rendering. We tested the proposed method to a new metropolitan sector which is currently under development and validated that the proposed rendering method is appropriate for the real-time rendering of complex huge scenes.

Acknowledgements. This work was supported in part by grant No. RTI05-03-01 from the Regional Technology Innovation Program of the Ministry of Commerce, Industry and Energy(MOCIE) and in part by the Brain Korea 21 Project in 2006.

References

1. Selçuk, A., Güdükbay, U., Özgüç, B.: A survey of interactive realistic walkthrough techniques in complex graphical environments. In: Advances in Computer and Information Sciences. (1998) 334–341
2. Chhugani, J., Purnomo, B., Krishnan, S., Cohen, J., Venkatasubramanian, S., Johnson, D., Kumar, S.: vlod: High-fidelity walkthrough of large virtual environments. IEEE Transactions on Visualization and Computer Graphics **11**(1) (2005) 35–47
3. Akenine-Möller, T., Haines, E.: Real-Time Rendering. A. K. Peters, Ltd., Natick, MA, USA (2002)
4. Samet, H., Webber, R.E.: Hierarchical data structures and algorithms for computer graphics. i. fundamentals. IEEE Computer Graphics and Application **8** (1988) 48–68
5. Pajarola, R.: Overview of quadtree-based terrain triangulation and visualization. Technical Report UCI-ICS TR 02-01, Dept. Info. Comp. Sci., University of California, Irvine (2002)
6. Wu, Y., Liu, Y., Zhan, S., Gao, X.: Efficient view-dependent rendering of terrains. In: GRIN'01: No description on Graphics interface 2001. (2001) 217–222
7. Pajarola, R.: Large scale terrain visualization using the restricted quadtree triangulation. In: Proceedings of IEEE Visualization'98. (1998) 19–24
8. Assarsson, U., Moller, T.: Optimized view frustum culling algorithms for bounding boxes. Journal of Graphics Tools **5**(1) (2000) 9–22
9. Clark, J.H.: Hierarchical geometric models for visible surface algorithms. Communications of the ACM **19**(10) (1976) 547–554
10. Lu, T., Chang, C.: Distributed visibility culling technique for complex scene rendering. Journal of Visual Languages and Computing **16** (2005) 455–479
11. Bittner, J., Wonka, P., Wimmer, M.: Visibility preprocessing for urban scenes using line space subdibision. In: Proceedings of Pacific Graphics (PG'01), IEEE Computer Society (2001) 276–284
12. Cohen-Or, D., Fibich, G., Halperin, D., Zadicario, E.: Conservative visibility and strong occlusion for viewspace partitioning of densely occluded scenes. Computer Graphics Forum **17**(3) (1998) 243–253
13. Falby, J.S., Zyda, M.J., Pratt, D.R., Mackey, R.L.: Npsnet: Hierarchical data structures for real-time three-dimensional visual simulation. Computers and Graphics **17**(1) (1993) 65–69
14. Gudukbay, U., Yilmaz, T.: Stereoscopic view-dependent visualization of terrain height fields. IEEE Transactions on Visualization and Computer Graphics **8**(4) (2002) 330–345
15. Yin, P., Shi, J.: Cluster based real-time rendering system for large terrain dataset. Computer Aided Design and Computer Graphics (2005) 365–370
16. Tsuji, T., Zha, H., Kurazume, R., Hasegawa, T.: Interactive rendering with lod control and occlusion culling based on polygon hierarchies. In: Proceedings of the Computer Graphics International (CGI'04). (2004) 536–539

Hierarchical Blur Identification from Severely Out-of-Focus Images

Jungsoo Lee[1], Yoonjong Yoo[1], Jeongho Shin[2], and Joonki Paik[1]

[1] Image Processing and Intelligent Systems Laboratory, Department of Image Engineering,
Graduate School of Advanced Imaging Science, Multimedia, and Film,
Chung-Ang University, Seoul, Korea
paikj@cau.ac.kr
http://ipis.cau.ac.kr
[2] Department of Web Information Engineering, Hankyong National University,
Ansung-city, Korea
shinj@hknu.ac.kr

Abstract. This paper proposes a blur identification method from severely out-of-focus images. The proposed blur identification algorithm can be used in digital auto-focusing and image restoration. Since it is not easy to estimate a point spread function (PSF) from severely out-of-focus images, a hierarchical approach is applied in the proposed algorithm. For severe out of focus blur, the proposed algorithm uses an hierarchical approach for estimating and selecting feasible PSF from successive down sampled images. The down sampled images contain more useful edge information for PSF estimation. The feasible PSF selected, can then be reconstructed for original image resolution level by up sampling methods. In order to reconstruct the PSF accurately, a regularized PSF reconstruction algorithm is used. Finally, we can restore the severely blurred image with the reconstructed PSF. Experimental results show that reconstructed PSF by the proposed hierarchical algorithm can efficiently restore severely out-of-focus images.

1 Introduction

There have been many researches for image restoration in the past decade [1], [5], [6], [7]. Due to the imperfection of practical imaging systems and inappropriate imaging conditions, the recorded images are often corrupted by degradations, such as blurring, noise and distortion, etc. Because of the bandwidth reduction of an image, blurring occurs in imaging system. In general, the degradation can be modeled as a linear convolution of original image with a point-spread function (PSF). The task from the linear modeling, image restoration can recover or estimate the original image from its degraded observation.

Image restoration can be classified into two categories such as direct restoration and iterative method. The direct restoration methods, such as inverse filtering, least-squares filters, etc. may fail to restore blurred images because of ill-posed inverse problem. Its solution is highly sensitive to noise or measurement errors. In practical applications, iterative restoration algorithms are often adopted, which have two main advantages over the direct algorithms [1], [4]. First, it can be terminated prior to convergence,

L.-W. Chang, W.-N. Lie, R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1334 – 1342, 2006.

resulting in a partially deblurred image which will often not exhibit noise amplification. The second advantage is that the inverse operator does not need to be implemented [1]. But iterative algorithms require high computational work.

The existing hierarchical techniques in [6], [7], however, could not provide acceptable performance due to difficulties in accurately estimating the blur parameters in severely blurred images. In this paper, we propose a new hierarchical blur identification and image restoration algorithm. We can estimate the PSF by analysis module based on boundary information from the image itself [2], [3]. Since it is hard to measure the edge information from severely blurred images, a pyramid-based hierarchical algorithm is used. After measuring PSFs from the edge of image pyramid, we use PSF reconstruction techniques to recover PSF with original support size.

The paper is organized as follows. In section 2, the hierarchical and image models are discussed; in section 3, the coefficients of PSF extraction scheme are presented; section 4 explains the regularized PSF reconstruction. Section 5 covers the experimental results and section 6 concludes the paper.

2 Hierarchical Model for Image Degradation

2.1 Image Degradation Model

A noisy blurred image is modeled as the output of a 2D linear space-invariant system, which is characterized by its impulse response. The output of a 2D linear system $g(m,n)$ is given as

$$g(m,n) = h(m,n) * f(m,n) + n(m,n) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} h(m-k, m-l) f(k,l) + n(m,n)\,, \tag{1}$$

where $h(m,n)$ represents the PSF that behaves like a smoothing operator, and $f(k,l)$ is the original input image with size $N \times N$. The last term, $n(m,n)$ represents an additive random noise under assumption that its distribution comply with zero-mean Gaussian density.

For further simplification (1) represents to use the matrix-vector notation

$$g = Hf + n\,, \tag{2}$$

Where f, g, and n are lexicographically ordered vectors of size $N^2 \times 1$, and H is the blurring operator of $N^2 \times N^2$.

2.2 Hierarchical Model

We assume that there is observed low-resolution PSF. The observation model is then

$$y_i = W_i g + \eta_i\,, \tag{3}$$

where g and y_i represent the lexicographically ordered desired high-resolution image and low-resolution image for i^{th} level. W_i and η_i represent the corresponding weights and additive Gaussian noise at level i . If we assume that the size of the PSF is $N \times N$, and then x is $N^2 \times 1$ vectors, y is $M^2 \times 1$ vectors and W is a $M^2 \times N^2$ matrix. From Eq. (2), a hierarchical model can be written as

$$y_i = W_i(Hf + n) + \eta_i = (\overline{H_i})(\overline{f_i}) + \overline{\eta}_i , \tag{4}$$

where $\overline{H_i}$, $\overline{f_i}$, and $\overline{\eta_i}$ represent $\overline{H_i} = W_i H$, $\overline{f_i} = W_i f$, and $\overline{\eta_i} = W_i n + \eta_i$, respectively.

3 Proposed PSF Identification Method

3.1 Hierarchical Structure

The prior PSF estimation method mentioned in section 3.2 requires that it has constraint; the number of prior PSF sets is finite. The fact, we can estimate more effective if support size of PSF is small or it has fewer PSF coefficients, implies that the support size of PSF decreases as well. Hence we propose hierarchical identification scheme.

The outline of hierarchical identification scheme is shown in Fig. 1. At first, we measure up to feasible edge information in the available blurred image in order to estimate PSFs using prior PSFs sets. If there is not enough edge information, then the blurred image is downsampled number of times by a factor of 2^i, denoted

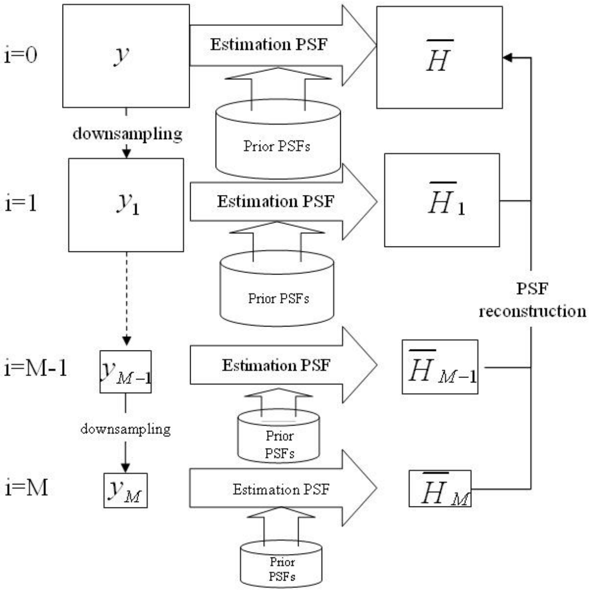


Fig. 1. Hierarchical blur identification procedure

by $\{y, y_1, y_2, \cdots, y_{M-1}, y_M\}$. By repeating the above process, the blurred image which is downsampled is measured to find feasible edge information. We identify $\overline{H_i}$ in Eq. (4). After identifying, we should do process that is PSF reconstruction to $\overline{H_1}$ from $\overline{H_i}$.

3.2 Prior PSF Estimation

The discrete approximation of an isotropic PSF is shown in Fig. 2. As shown in Fig. 2, many pixels are located off concentric circles within the region defined as

$$S_R = \{ (m,n) \mid \sqrt{m^2+n^2} \le R \}, \tag{5}$$

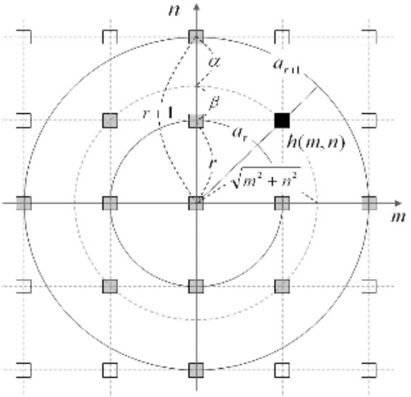


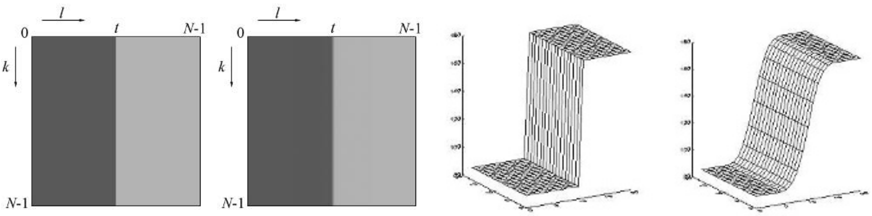
Fig. 2. The geometrical representation of a 2D isotropic discrete PSF: The gray rectangles represent inner pixels of the PSF and the empty ones are outer pixels of the PSF

where R is the radius of the PSF. Each pixel within the support S_R is located either on the concentric circles or not. The pixels on a concentric circle are straightforwardly represented as the PSF coefficients, $a_0, \cdots, a_R$, as shown in Fig. 2. On the other hand, pixels off a concentric circle are not described by those. So these pixels should be interpolated by using adjacent pixel on the concentric circle as

$$\tilde{h}(m,n) = \begin{cases} \alpha a_r + \beta a_{r+1}, & \text{if } (m,n) \in S_R, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where a_r and a_{r+1} respectively represent the rth and the $r+1$st entries of the PSF coefficient vector. In (6), index r is determined as

$$r = \left\lfloor \sqrt{m^2+n^2} \right\rfloor, \tag{7}$$



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

Fig. 3. The pattern images and its edges: (a) original simple pattern image, (b) blurred pattern image, (c) edges of (a), and (d) edges of (b)

where $\lfloor \cdot \rfloor$ is the truncation operator to integer. Based on Fig. 2, α and β are determined as

$$\alpha = r + 1 - \sqrt{m^2 + n^2} \text{ , and } \beta = 1 - \alpha. \tag{8}$$

This approximation of 2D discrete PSF is available to the popular isotropic blurs, such as Gaussian out-of-focus blur, uniform out-of-focus blur, and x-ray scattering. The optimal PSF's were selected using the out-of-focus estimate by calculating the distance between the unit step function and a luminance profile across the edge of the restored with various PSF's as

$$d^{(k)} = \left\| s^{(k)} - u \right\|_2 = \left(\sum_{j=1}^{N} \left| s_j^{(k)} - u_j^{\,2} \right| \right)^{1/2} \text{, for } k = 1, 2, \cdots, K., \tag{9}$$

where $s^{(k)}$ represents the restored edge profile with the k^{th} pre-estimated PSF in the selected auto-focusing region, and u the unit step function. Here, N denotes the number of points in the edge profile and k is an index for a PSF in the set of K members. The optimal PSF can be determined by choosing the k^{th} PSF with minimum distance as

$$\arg\min \{ d^{(k)} \}, \text{ for } k = 1, 2, \cdots, K. \tag{10}$$

4 PSF Reconstruction Using Spatially Adaptive Constraints

In this section, we mention how the estimated PSF for down sampled images can be used to overcome the out of focus blur at higher resolution images. The estimated PSF is extended and generalized to be applied to its corresponding higher resolution images.

4.1 PSF Reconstruction for High Resolution Restoration

The PSF reconstruction problem can be started as the problem of finding the inverse of the down-sampling operator W_i in Eq. (3) or finding an PSF which is as close as possible to the original one, subject to an optimality criterion. The PSF reconstruction problem is as ill-posed problem, which results in the matrix W_i being ill-conditioned. A regularization approach is required in solving ill-posed problems. Such an approach controls the noise amplification in the solution by incorporation a priori information about the solution into the restoration process, and it results in the minimization with respect to H of the functional

$$\Phi(x) = (\left\|\bar{H}_i - W_i H\right\|^2 + \lambda\left\|CH\right\|^2) \uparrow 2 \tag{11}$$

The functional $\left\|CH\right\|$ is called the stabilizing function since it bounds the energy of the restored signal at high frequencies, primarily due to the amplified noise. By minimizing the first term of $\Phi(\lambda = 0)$, the solution vector H tends to $W_i^{-1}\bar{H}_i$, resents the smoothest possible solution. The tradeoff between deconvolution and noise smoothing operation is determined by the parameter λ.

By neglecting the constant terms with respect to the variable H in Eq. (11), the minimization of $\Phi(H)$ is equivalent to the minimization of $f(H)$, where

$$f(H) = \frac{1}{2} H^T T H - b^T H \tag{12}$$

and

$$T = W_i^T W_i + \lambda C^T C \text{ and } b = W_i^T \bar{H}_i . \tag{13}$$

The necessary condition for a minimum of $f(H)$ result in the solution of the system of linear equation

$$TH = b \tag{14}$$

Iterative methods are used in this work in solving Eq. (14).

Among the advantages of iterative algorithms is the possibility of incorporation constraints, expressing properties of the solution into the PSF reconstruction process. The regularized solution given by (14) can be successively approximated by means of the iteration

$$\begin{gathered} H_0 = 0, \\ H^{n+1} = H^n + \beta(W_i^T \bar{H}_i - W_i^T W_i H^n - \lambda(H^n) C^T C H^n), \end{gathered} \tag{15}$$

where β and $\lambda(H)$ represent a multiplier that can be constant or it can depend on the iteration index and regularization functional.

4.2 Regularization Function

We are interested to seek to incorporate the following desirable properties for the regularization function at each iteration step. We further simplify the analysis by assuming in [4] that ⅰ) $\lambda(H)$ is inversely proportional to $\left\|\bar{H}-WH\right\|^2$, ⅱ) $\lambda(H)$ is proportional to $\left\|CH\right\|^2$, ⅲ) $\lambda(H)$ is larger than zero. Having all there assumptions, we have the following regularization function according to the properties described above.

$$\lambda(H)=\theta(\frac{\left\|CH\right\|^2}{\left\|\bar{H}-WH\right\|^2+\delta}), \tag{16}$$

where θ is a monotonically increasing function. δ prevents the denominator from becoming zero. In this paper, Eq. (16) is defined by

$$\lambda(H)=\ln(\frac{\left\|CH\right\|^2}{\left\|\bar{H}-WH\right\|^2+\delta}). \tag{17}$$

5 Experimental Results

In this section, we present result that is dealt with the synthetically blurred images by two different of PSFs.

Experiment I: In this experiment, Fig. 4(a) shows the original image, 'man', with the size of 512×512 pixels. We synthetically make an out-of-focused version this image by convolving with 17×17 Gaussian that the variance of Gaussian function, $\sigma = 3$, is used. 30dB blurred signal-to-noise ratio (BSNR) noise is added after the simulated out-of-focus blur. The degraded version of the 'man' is shown in Fig. 4(b). As the first step of the proposed algorithm, the feasible edges should be extracted from the image shown in Fig. 4(b) according to the algorithm described in section 3.2. This picture, however, has the lack of feasible edge information. Thus, the original image is downsampled to level 2 by a factor of 4, as shown in Fig. 4(c). Fig. 4(d) shows the restored image of downsampling from Fig. 4(c) using PSF estimation by prior PSF method. The set of prior PSFs are Gaussian blur that support size is 3,5 and 7 variance of Gaussian is between 0.04~4.04, and the number of these is 300. Finally, Fig. 4(e) shows the restored image has the original support size by PSF construction described in section 4. We use a constraint least squares (CLS) filter to restore the out-of-focus image.

Experiment II: Synthetically, uniform out-of-focus blur with 9×9 is also used for another experiment. 35dB BNSR additive is further added to the blurred image. The original image is 'peppers' with 512×512 pixels in Fig. 5(a). Fig. 5(b) shows degraded image with 9×9 uniform blur. Fig. 5(c) shows downsampling image at downsampled level 1 by a factor of 2. Fig. 5(d) shows the restored image of Fig. 5(c) using prior PSF estimation. The set of prior PSFs is the same set used in experiment I. Fig. 5 (e) shows the restored image using CLS filter based on proposed algorithm.

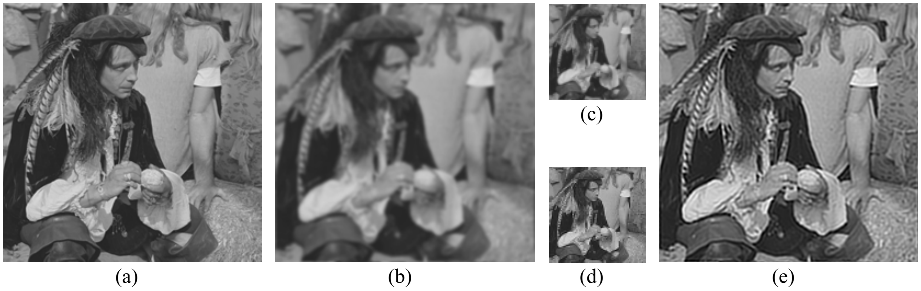


Fig. 4. The experimental results of experiment I: (a) original Image, (b) the degraded Image, (c) downsampled image from (b) at level 2, (d) the restored image of downsampling from (c), and (e) the restored image by the proposed algorithm

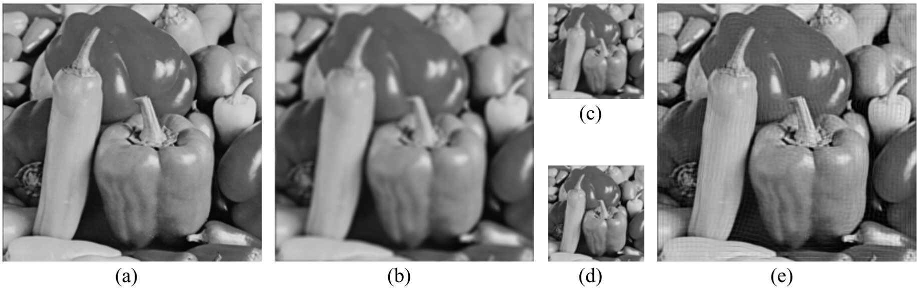


Fig. 5. The experimental results of experiment II: (a) original Image, (b) the degraded Image, (c) downsampled image from (b), (d) the restored image of downsampling from (c) at level 1, and (e) the restored image by the proposed algorithm

6 Conclusions

In this paper, we have proposed a hierarchical PSF estimation and image restoration algorithm for severely blurred image by using a prior PSF estimation approach based on feasible edge information. We identify the PSF having a large support size. This algorithm providing the optimum result decreased. The proposed algorithm is robust against the large support size of blur. Experimental results showed that high-frequency details can be restored from the severely blurred images. Although the proposed algorithm is currently limited to space invariant, isotropic PSFs, it can be extended to more general applications based PSF estimation.

Acknowledgment

This research is supported by Korean Ministry of Science and Technology under National Research Laboratory Project and by Korean Film Council under Culture Technology Development Project.

References

1. J. Biemond, L. L. Reginald, and R. M. Mersereau, “Iterative Methods for Image Deblurring,” IEEE Trans. on Image Processing, vol. 78, no. 5, pp. 856-883, May 1990
2. S. K. Kim, S. R. Paik, and J. K. Paik, “Simultaneous Out-of-Focus Blur Estimation and Restoration for Digital AF System,” IEEE Trans. Consumer Electronics, vol. 44, no. 3, pp. 1071-1075, August 1998.
3. S. K. Kim and J. K. Paik, “Out-of-Focus Blur Estimation and Restoration for Digital Auto-Focusing System,” Electronics letters, vol. 34, no. 12, pp. 1217-1219, June 1998.
4. E. S. Lee and M. G. Moon, “Regularized Adaptive high-Resolution Image Reconstruction Considering Inaccurate Subpixel Registration,” IEEE Trans. on Image Processing, vol. 12, no. 7, July 2003.
5. A. K. Katsaggelos, “Iterative Image Restoration Algorithms,” Optical Engineering, vol. 28, no. 7, pp. 735-748, July 1989.
6. R. L. Lagendijk, J. Biemond and D. E. Boekee, “Hierarchical Blur Identification,” Acoustics, Speech, and Signal Processing, vol. 4, pp. 1889-1892, April 1990.
7. N. P. Galatsanos, V. Z. Mesarovic, R. Molina, and A. K. Katsaggelos, “Hierarchical Bayesian Image Restoration from Partially Known Blurs,” IEEE Trans. on Image Processing, vol. 9, no. 10, pp. 1784 – 1797, Oct 2000.

Author Index