LNCS 7087

Yo-Sung Ho (Ed.)

Advances in Image and Video Technology

5th Pacific Rim Symposium, PSIVT 2011
Gwangju, South Korea, November 2011
Proceedings, Part I

1

Part I

Springer

Lecture Notes in Computer Science 7087

Yo-Sung Ho (Ed.)

Advances in Image and Video Technology

5th Pacific Rim Symposium, PSIVT 2011
Gwangju, South Korea, November 20-23, 2011
Proceedings, Part I

Springer

Volume Editor

Yo-Sung Ho
Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong Buk-gu, Gwangju, 500-712, South Korea
E-mail: hoyo@gist.ac.kr

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-25366-9 e-ISBN 978-3-642-25367-6
DOI 10.1007/978-3-642-25367-6
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011940679

CR Subject Classification (1998): H.5.1, H.5, I.4-5, I.2.10, I.3, H.3-4, E.4

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

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

We are delighted to welcome readers to the proceedings of the 5th Pacific-Rim Symposium on Video and Image Technology (PSIVT 2011), held in Gwangju, Korea, during November 20-23, 2011. The first PSIVT was held in Hsinchu, Taiwan, in 2006. Since then, it has been hosted successfully by Santiago, Chile, in 2007, Tokyo, Japan, in 2009, Singapore in 2010, and finally Gwangju, one of the beautiful and democratic cities in Korea. The symposium provides a forum for presenting and discussing the latest research and development in image and video technology and explores possibilities and future directions in the field. PSIVT 2011 continued to attract researchers, artists, developers, educators, performers, and practitioners of image and video technology from the Pacific rim and around the world.

In PSIVT 2011, the Program Committee was made up of Area Chairs and a Technical Program Committee. The technical areas of PSIVT 2011 covered Image/Video Coding and Transmission, Image/Video Processing and Analysis, Imaging and Graphics Hardware and Visualization, Image/Video Retrieval and Scene Understanding, Biomedical Image Processing and Analysis, Biometrics and Image Forensics, and Computer Vision Applications. For each technical area, at least two Area Chairs were assigned to coordinate the paper-review process with their own team of reviewers selected from the Technical Program Committee. The review process was double-blind in which author names and affiliations were not made known to Area Chairs and reviewers. Reviewers also did not know their Area Chairs. Each paper received at least three reviews. The reviewers were asked to submit a detailed review report and the Area Chairs made the final decisions on the acceptance of papers with little moderation from the Program Chairs. In PSIVT 2011, we accepted 71 papers out of 168 submissions including oral and poster session papers. The acceptance rate of 42% indicates our commitment to ensuring a very high-quality symposium.

PSIVT 2011 was organized by the Realistic Broadcasting Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST) in Korea. The symposium was supported by the Center for Information Technology Education (BK21) at GIST, Gwangju Convention and Visitors Bureau, and the MPEG Forum in Korea.

This symposium would not be possible without the efforts of many people. First of all, we are very grateful to all the authors who contributed their high-quality research work and shared their knowledge with our scientific community. We would also like to appreciate the full support of the excellent Program

Committee and all reviewers that provided timely and insightful reviews. Finally, our thanks must go to all members of the Organizing and Steering Committee for their precious time and enthusiasm. They did their best in financing, publicity, publication, registration, Web and local arrangements.

November 2011 Yo-Sung Ho

PSIVT 2011 Organization

Organizing Committee

General Co-chairs

Yo-Sung Ho	Gwangju Institute of Science and Technology, Korea
Wen-Nung Lie	National Chung Cheng University, Taiwan
Domingo Mery	Pontificia Universidad Catolica, Chile

Program Co-chairs

Kap Luk Chan	Nanyang Technological University, Singapore
Qingming Huang	Chinese Academy of Sciences, China
Shin'ichi Satoh	National Institute of Informatics, Japan

Finance Chair

Kuk-Jin Yoon	Gwangju Institute of Science and Technology, Korea

Publicity Co-chairs

Sung-Hee Lee	Gwangju Institute of Science and Technology, Korea
Yousun Kang	Tokyo Polytechnic University, Japan

Publication Chair

Sung Chan Jun	Gwangju Institute of Science and Technology, Korea

Local Arrangements Chair

Hyunju Lee	Gwangju Institute of Science and Technology, Korea

Steering Committee

Kap Luk Chan	Nanyang Technological University, Singapore
Yung-Chang Chen	National Tsinghua University, Taiwan
Yo-Sung Ho	Gwangju Institute of Science and Technology, Korea
Reinhard Klette	The University of Auckland, New Zealand
Wen-Nung Lie	National Chung Cheng University, Taiwan
Domingo Mery	Pontificia Universidad Catolica, Chile
Akihiro Sugimoto	National Institute of Informatics, Japan
Mohan M. Trivedi	University of California, San Diego, USA

Area Chairs

Oscar Au	Hong Kong University of Science and Technology, Hong Kong
Miguel Carrasco	Universidad Diego Portales, Chile
Yoong Choon Chang	Multimedia University, Malaysia
Anthony TS Ho	University of Surrey, UK
Fay Huang	National Ilan University, Taiwan
Shuaqiang Jiang	Chinese Academy of Sciences, China
Shang-Hong Lai	National Tsing Hua University, Taiwan
Jaejoon Lee	Samsung Electronics, Korea
Qingshan Liu	Rutgers University, USA
Chia-Wen Lin	National Tsing-Hua University, Taiwan
Huei-Yung Lin	National Chung Cheng Uiversity, Taiwan
Yasuhiro Mukaigawa	Osaka University, Japan
Luis Pizarro	Imperial College, UK
Mingli Song	Zhejiang University, China
Yu-Wing Tai	KAIST, Korea
Gang Wang	Nanyang Technological University, Singapore
Lei Wang	University of Wollongong, Australia
Changsheng Xu	Chinese Academy of Sciences, China
Shuicheng Yan	National University of Singapore, Singapore
Junsong Yuan	Nanyang Technological University, Singapore
Jianxin Wu	Nanyang Technological University, Singapore
Vitali Zagorodnov	Nanyang Technological University, Singapore

Technical Program Committee

Hezerul Abdul Karim
Toshiyuki Amano
Yasuo Ariki
Vishnu Monn Baskaran
Bedrich Benes
Xiujuan Chai
Yoong Choon Chang
Chin-Chen Chang
Chia-Yen Chen
Yi-Ling Chen
Chu-Song Chen
Jia Chen
Hwann-Tzong Chen
Jian Cheng
Gene Cheung
Chen-Kuo Chiang
Sunghyun Cho
Michael Cree
Ismael Daribo
Xiaoyu Deng
Lei Ding
Zhao Dong
Gianfranco Doretto
How-Lung Eng
Giovani Gomez
Gerardo Fernández-Escribano
Chiou-Shann Fuh
Makoto Fujimura
Hironobu Fujyoshi
Kazuhiro Fukui
Simon Hermann
Yo-Sung Ho
Seiji Hotta
Jun-Wei Hsieh

Changbo Hu
Xiaoqin Huang
Rui Huang
Junzhou Huang
Chun-Rong Huang
Naoyuki Ichimura
Masahiro Iwahashi
Daisuke Iwai
Yoshio Iwai
Gangyi Jiang
Xin Jin
Ramakrishna Kakarala
Masayuki Kanbara
Li-Wei Kang
Hiroshi Kawasaki
Chang-Su Kim
Itaru Kitahara
Mario Koeppen
Akira Kubota
Takio Kurita
Shang-Hong Lai
Tung-Ying Lee
Wen-Nung Lie
Chia-Wen Lin
Guo-Shiang Lin
Xiao Liu
Damon Shing-Min Liu
Huiying Liu
Jonathan Loo
Yasushi Makihara
Takeshi Masuda
Fabrice Meriadeau
Rodrigo Moreno
Hajime Nagahara
Atsushi Nakazawa
Kai Ni
Shohei Nobuhara
Takeshi Oishi
Takahiro Okabe
Ho-Yuen Pang
Christian Pieringer
Lei Qin
Bo Qiu
Mauricio Reyes
Laurent Risser
Isaac Rudomin
Clarisa Sanchez
Tomokazu Sato
Takeshi Shakunaga
Shiguang Shan
Xiaowei Shao
Chunhua Shen
Ikuko Shimizu
Keita Takahashi
Toru Tamaki
Ping Tan
Masayuki Tanaka
Flavio Torres
Chien-Cheng Tseng
Seiichi Uchida
Carlos Vazquez
Yu-Chiang Wang
Jingqiao Wang
Min-Liang Wang
Hsien-Huang Wu
Ming Yang
Chia-Hung Yeh
Kaori Yoshida
Guangtao Zhai
Daoqiang Zhang
Qi Zhao
Yuanjie Zheng
Bo Zheng
Huiyu Zhou
Shaohua Zhou

Sponsoring Institutions

The Realistic Broadcasting Research Center (RBRC) at GIST
The Center for Information Technology Education (BK21) at GIST
Gwangju Convention and Visitors Bureau
The MPEG Forum in Korea

Table of Contents – Part I

Table of Contents – Part II

Nonlinear Transfer Function-Based Image Detail Preserving Dynamic Range Compression for Color Image Enhancement

Deepak Ghimire and Joonwhoan Lee

Computer Science and Engineering, Chonbuk National University,
Jeonju 561-756, Jeollabuk-do, Rep. of Korea
{deep,chlee}@chonbuk.ac.kr

Abstract. This paper presents a method for color image enhancement in HSV space with preserving image details. The RGB color image is converted into HSV space and V channel image is now subjected for enhancement. By applying image dependent nonlinear transfer function the local image contrast preserving dynamic range compression as well as contrast enhancement is performed simultaneously on the V channel image. Finally, the enhanced V channel image and original H and S channel images are converted back to RGB image to obtain enhanced RGB image. The original color of the image is preserved because H and S component are kept unchanged. The experimental results show that the performance of the proposed method is better in terms of both subjective and objective evaluation in comparison with conventional methods.

Keywords: nonlinear transfer function, dynamic range compression, image local contrast, multiscale enhancement.

1 Introduction

The image captured in natural environment with high dynamic range (HDR) includes both bright and dark regions. The camera has the capability to capture high dynamic range images, while most of the display devices have low dynamic range. On the other hand if the dynamic range of human eye sensing is exceeded, it is difficult to perceive the HDR images. Image enhancement with dynamic range compression is a common approach to improve the quality of those images in terms of human visual perception. The image enhancement techniques can be divided into two categories namely: spatial domain methods and transform domain methods. In spatial domain methods the intensity of the pixel in image is directly manipulated. But in transform domain techniques the image intensity data is transformed into specific domain by using methods such as DFT, DCT, DWT, etc. and the frequency content of the image is altered for image enhancement.

Various image processing technique have been developed to improve the quality of the image in terms of human visual perception. One of the traditional and well known techniques for image contrast enhancement is histogram equalization (HE). But HE in its original form tends to introduce some annoying artifacts and unnatural enhancement. To overcome those problems different variants of traditional HE method are developed.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 1–12, 2011.

In the literature methods like [1], [2], [3], [4] etc. can be found as modified HE based method for image contrast enhancement. Methods like [1], [2] divides image into two sub images based on mean and median of the original image respectively and performs the HE in each image independently. At last, the processed sub images are composed into one image to get the final result. An improved version of those methods is presented in [3]. Here separation is done recursively; separates each new histogram further based on their respective mean. As the number of separation increases, the output image's mean brightness will converge to the input image's mean brightness. Similarly method [4] modifies the histogram of the original image by weighting and thresholding before the HE. But, the problem with HE based methods is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal. On the other hand it produces unrealistic effects in the images.

R. Fattel et al. [5] developed a gradient domain high dynamic range compression method. They modified the gradient field of the luminance image by attenuating magnitude of the large gradients and obtain the low dynamic range image by solving a Poisson equation on the modified gradient field. On the other hand Debevec et al. [6] develop a method of recovering high dynamic range radiance maps from ordinary photographs. Their algorithm used the multiple differently exposed photographs to recover the response function of the imaging process and with this response function the algorithm can fuse the multiple photographs into single, high dynamic range radiance map. The multiscale retinex based method, e.g., Jobson et al. [7], and single scale retinex based image enhancement method, e.g., Choi et al. [8] are also developed, in which luminance enhancement and contrast enhancement are realized simultaneously. Tao et al. [9] presented an adaptive and integrated neighborhood dependent approach for nonlinear enhancement (AINDANE) to improve the visual quality of digital images captured under low or nonuniform illumination conditions. It consists of two independent processes: adaptive luminance enhancement and adaptive contrast enhancement, which are applied to treat luminance information of images. Recently, many transform based enhancement techniques have also been developed. Xiao et al. [10] proposed a method for enhancing contrast of the image based on wavelet transform and human visual system. Clement et al. [11] developed the image enhancement algorithm in compressed DCT domain which is able to enhance both dark and bright region of an image equally well.

Most of the digital video cameras have adopted a knee curve as a dynamic range compression function. This method strongly compressed the highlighted range over the knee point, so the contrast in the highlighted region is much degraded. To improve this problem an auto knee curve has been used for dynamic range compression. However, the auto knee slightly improves the highlight contrast instead of lowering the luminance in the middle range. To solve these problems an approximated auto knee curve is used by Monobe et al. [12] for dynamic range compression with preserving local image contrast. This algorithm automatically and adaptively enhances the local image contrast in the highlighted regions. Here, because the use of approximated auto knee curve local contrast capabilities are limited. On the other hand, we would like to able to have strong enhancement capability and like to extend the enhancement in the middle and low frequency regions. To achieve this goal, in this paper we are using the concept of contrast preserving proposed by [12] with nonlinear transfer function as dynamic range compression function. In this paper the concepts from [9] and [12] are combined to achieve color image enhancement in HSV space. Image enhancement in HSV space has the advantage of preserving color and saturation of the image by only modifying value component of the original image in the enhancement process. Recently, D. Ghimire et al. [13] presented a method for image enhancement in HSV space by considering image locality in dynamic range compression process. This is the improved version of the method proposed in [9]. In this paper we are also using the concept of multiscale image convolution to improve the result of image enhancement.

The rest of the paper is organized as follows. In section 2, the procedure of image enhancement in HSV space is presented. In section 3, experimental results along with performance evaluation of the proposed method are shown. Finally the conclusion is given in section 4.

2 Local Contrast Preserving Image Enhancement

In general, color images are represented in RGB color space. This paper uses HSV color space for image enhancement, in which the hue (*H*) refers to the spectral composition of color, saturation (*S*) defines the purity of colors and value (*V*) refers the brightness of a color or just the luminance value of the color. Here RGB values of the image are converted into HSV values and then the value component image is now subjected for enhancement. HSV color space is selected in this image enhancement procedure to preserve the saturation and color of the input image.

In this paper we are using the basic concept of local contrast preserving proposed in [12] for dynamic range compression. The mathematical condition to preserve the local contrast is described as follows.

$$\frac{g(x,y)}{g_{ave}(x,y)}=\frac{f(x,y)}{f_{ave}(x,y)} \tag{1}$$

where, $f(x, y)$ and $f_{ave}(x, y)$ denotes the input luminance level and the input local average, $g(x, y)$ and $g_{ave}(x, y)$ denotes the output luminance level and the output local average of each pixel (x, y) of the value component image in HSV space respectively.

The core equation describing the condition to preserve the local contrast in dynamic range compression process given by [12] is described as follows.

$$g(x,y)=p(f(x,y))\times r(f(x,y),f_{ave}(x,y)) \tag{2}$$

$$r(f(x,y),f_{ave}(x,y))=\left(\frac{f(x,y)}{f_{ave}(x,y)}\right)^{\alpha\left\{1-\frac{f(x,y)}{p(f(x,y))}\frac{dp(f(x,y))}{df(x,y)}\right\}} \tag{3}$$

where, $p(f(x, y))$ denotes an arbitrary tone mapping curve in luminance domain, and α denotes a gain parameter for the local contrast enhancement.

The local average $f_{ave}(x, y)$ in (2) and (3) is calculated by taking the convolution of spatial averaging filter $A(x, y)$ and the input value component image $f(x, y)$ as follows,

$$f_{ave}(x,y)=A(x,y)\otimes f(x,y) \tag{4}$$

Here, the 2-D Gaussian function $A(x, y)$ is defined as

$$A(x,y) = K\exp\left[\frac{-(x^2+y^2)}{\sigma^2}\right] \tag{5}$$

The standard deviation (σ) of the 2-D Gaussian distribution determines the size of the neighborhood. In this equation K is a gain factor and is determined by

$$\iint K\exp\left[\frac{-(x^2+y^2)}{\sigma^2}\right]dx\,dy = 1 \tag{6}$$

The selection of tone mapping function, $p(f(x, y))$ in Eq. (2) and (3) is very important and will affect the result of enhancement directly. In [12], the authors are using approximated knee curve as a tone mapping curve. But the knee curve only compresses the highlighted range over the knee point. On the other hand here we want the strong enhancement capabilities and like to extend the enhancement in the middle and low frequency regions too. Different type of tone mapping function can be used to have different type of enhancement results. In this paper we are trying to enhance the regions with low intensity or with dark pixels. Therefore, we have selected the nonlinear transfer function as mapping function used by [9] for luminance enhancement which also serves as dynamic range compression and is defined as

$$p(f(x,y)) = \frac{f(x,y)^{(0.75z+0.25)} + 0.5(1-f(x,y))(1-z) + f(x,y)^{(2-z)}}{2} \tag{7}$$

The range of $f(x, y)$ in Eq. (7) is 0 to 1. The nonlinear transfer function provided in Eq. (7) is image dependent with parameter z and is calculated by using following relation

$$z = \begin{cases} 0 & \text{for } L \le 50 \\ \dfrac{L-50}{100} & \text{for } 50 < L \le 150 \\ 1 & \text{for } L > 150 \end{cases} \tag{8}$$

where L is the luminance level corresponding to a cumulative distribution function (CDF) of 0.1 of value component input image in HSV space. The range of L is 0 to 255.

The image dependent parameter z can be calculated from the image globally, or can be calculated from the value component image locally. Calculating parameter locally depending upon each small region of the image and applying luminance enhancement with different shaped transfer function for each corresponding small region can produce better result than applying global transfer function in luminance enhancement process [13]. But, calculating parameter locally increases the computational complexicity of the algorithm. Fig. 1 shows the shape of nonlinear transfer function for different values of z and Fig. 2 shows an input value component image along with corresponding CDF.

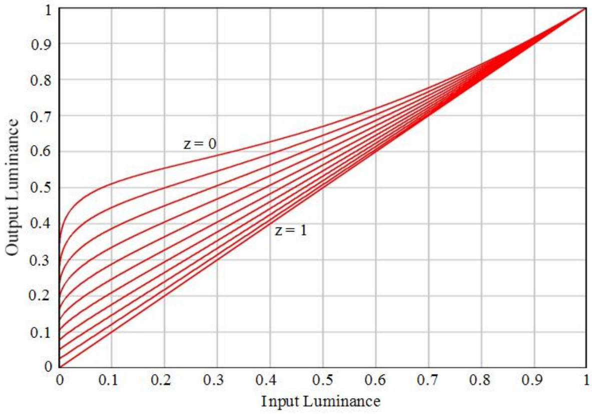


Fig. 1. Shape of nonlinear transfer functions with different z values

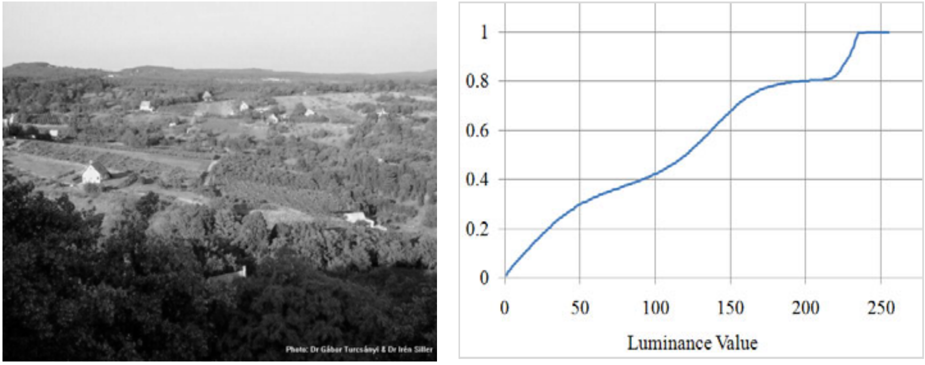


Fig. 2. An intensity image and its cumulative distribution function (CDF)

According to Eq. (3) we need differential function of nonlinear transfer function of Eq. (7) which is defined as

$$\frac{dp(f(x,y))}{df(x,y)} = \frac{(0.75z+0.25)f(x,y)^{(0.75z-0.75)}+0.5(z-1)+(2-z)f(x,y)^{(1-z)}}{2} \tag{9}$$

Now Eq. (3) can be applied for image enhancement. In our experiment we are using value of α as 1.5, because we want to increase the local contrast of the input image. Selection of scale (standard deviation) in Gaussian function is another important aspect in image enhancement procedure because it directly affects the result of enhancement. Convolution with a small scale, such as a few neighboring pixels, can provide luminance information about the nearest neighborhood pixels, while the convolution with a large scale comparable to the image dimensions can provide the information about the large-scale luminance variation over the whole image. Generally, smaller scale convolution tends to produce result with fine details and convolution with larger scale tends to produce natural looking and smooth results. A medium scale convolution can produce combination of both small scale and large scale results. Therefore we can use multiple scale convolutions to produce different results and we can combine all of them to find the final image enhancement result. But, if we need faster processing we can use medium scale convolution in image enhancement process. On the other hand we can also use parallel processing to find the enhancement results in different scales to get rid from the computational complexicity. The image enhancement with multiscale convolution can be described by the following equations

$$g_i(x,y) = p(f(x,y)) \times r(f(x,y), f_{ave,i}(x,y)) \tag{10}$$

$$r(f(x,y), f_{ave,i}(x,y)) = \left(\frac{f(x,y)}{f_{ave,i}(x,y)} \right)^{\alpha \left\{ 1 - \frac{f(x,y)}{p(f(x,y))} \frac{dp(f(x,y))}{df(x,y)} \right\}} \tag{11}$$

$$f_{ave,i}(x,y) = A_i(x,y) \otimes f(x,y) \tag{12}$$

$$A_i(x,y) = K \exp\left[\frac{-(x^2 + y^2)}{\sigma_i^2} \right] \tag{13}$$

$$g(x,y) = \frac{\sum_{i=1}^{n} g_i(x,y)}{n} \tag{14}$$

where n is the number of scales and σ_i represents different scales.

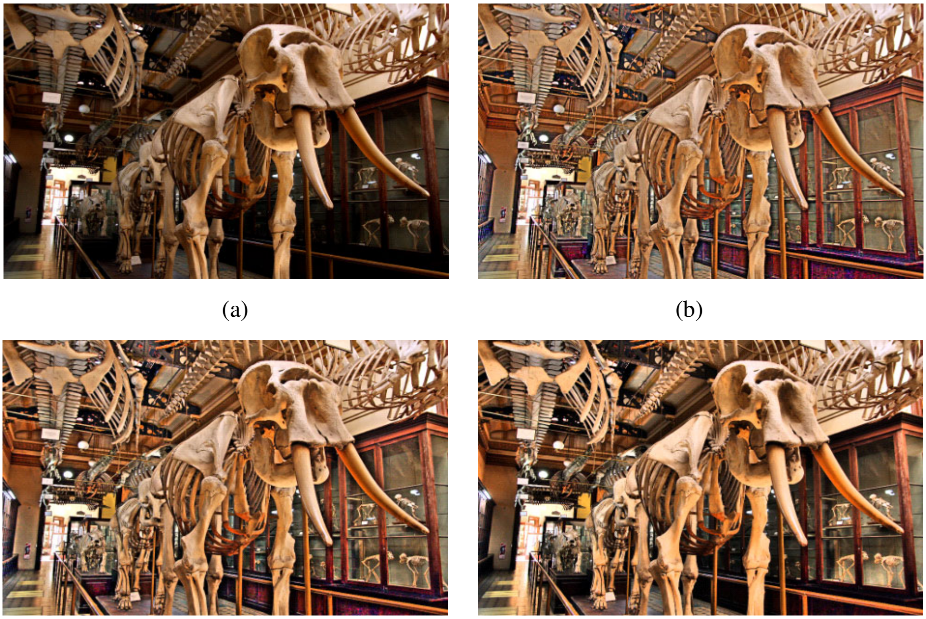
In our experiment, we selected $n = 3$ and find the enhanced images in three scales: small, medium and large. In this work, used three scales are 5, 15 and 50. It is experimentally determined that those scales are suitable for almost all type of images. After obtaining the average enhancement result of value component image in HSV space by using Eq. (14), it is combined with original *H* and *S* component images and converted back to RGB space to find the final result in RGB space. The saturation and hue channels are not altered in this image enhancement procedure.

3 Results and Discussion

The proposed algorithm has been applied to large number of digital images captured under dark illumination conditions for performance evaluation and comparison with other techniques. This section contains some results as well as discussion about the performance of the proposed algorithm for image enhancement. The proposed algorithm is also compared with other methods using both subjective and objective evaluation.

3.1 Subjective Evaluation

Image enhanced with various scales in Gaussian function are shown in Fig. 3. The effect of different scales can be clearly seen in these enhanced images. Result of image enhancement using very small scale has the richer local details and result by using large sale has smoothing effect. Convolution with medium scale has the result in somewhere middle of both large and small scale result. Therefore, from this result it is clear that combination of all the scales can produce better results. But the computational complexity is increased in multiscale convolution image enhancement process if computation is carried out in serial manner.



(a) (b)

(c) (d)

Fig. 3. Image enhancement results with different scales in Gaussian function: (a) Input image, (b) enhancement using scale = 5, (c) enhancement using scale = 10, and (d) enhancement using scale = 50

Here we compare the performance of the proposed method with the HE, AINDANE [9], MSRCR [7] and literature [13]. The enhanced image using proposed method has fine local and global details with natural looking, and balanced luminance and contrast across the whole image and no change in original color of the image in comparison with other methods. Fig. 4 shows two input color images and result of enhancement using different methods. The color of the image is changed as well as the local contrast of the image is increased unnecessarily using MSRCR method. The global enhancement using AINDANE and [13] are satisfactory, but local details are still to be enhanced. In the output image of our method both local and global contrast are increased well with preserving image details. Here we are using miltiscale convolution for enhancement using proposed method. More results of image enhancement using proposed method are shown in Fig. 5. The experiment on other test images has shown similar results.

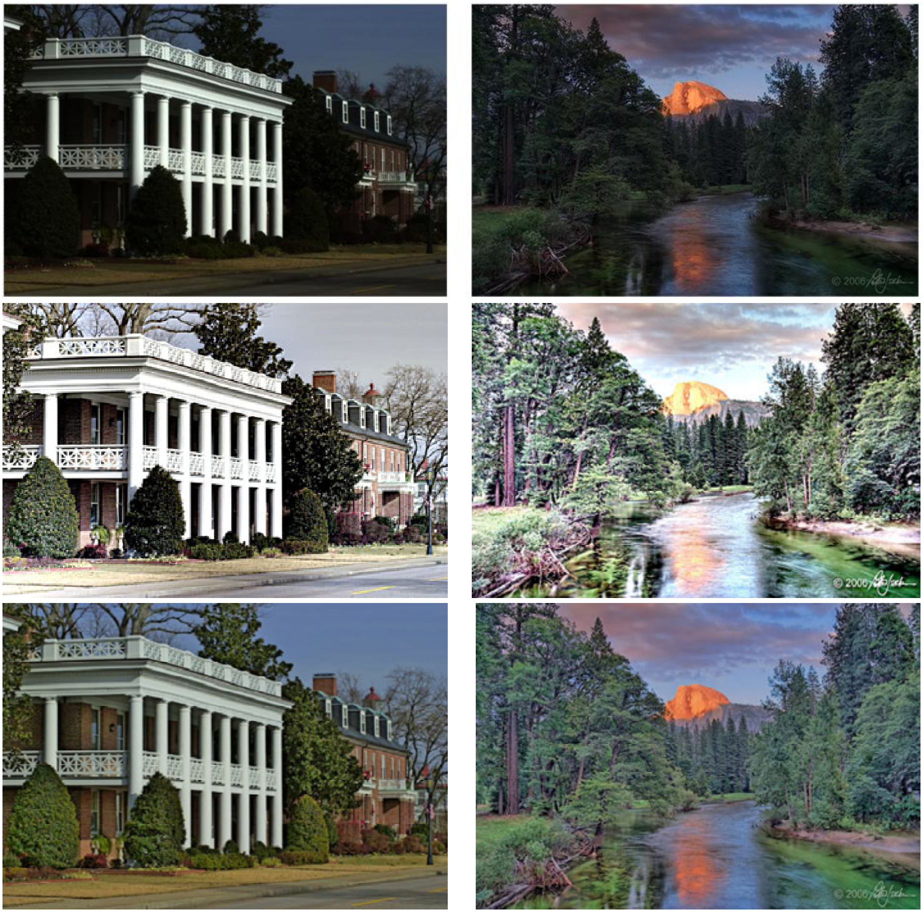


Fig. 4. Comparison of image enhancement with different methods: Input color images (first row), enhancement results using MSRCR (second row), enhancement results using AINDANE (third row), enhancement results using [13] (fourth row) and enhancement results using proposed method (last row)

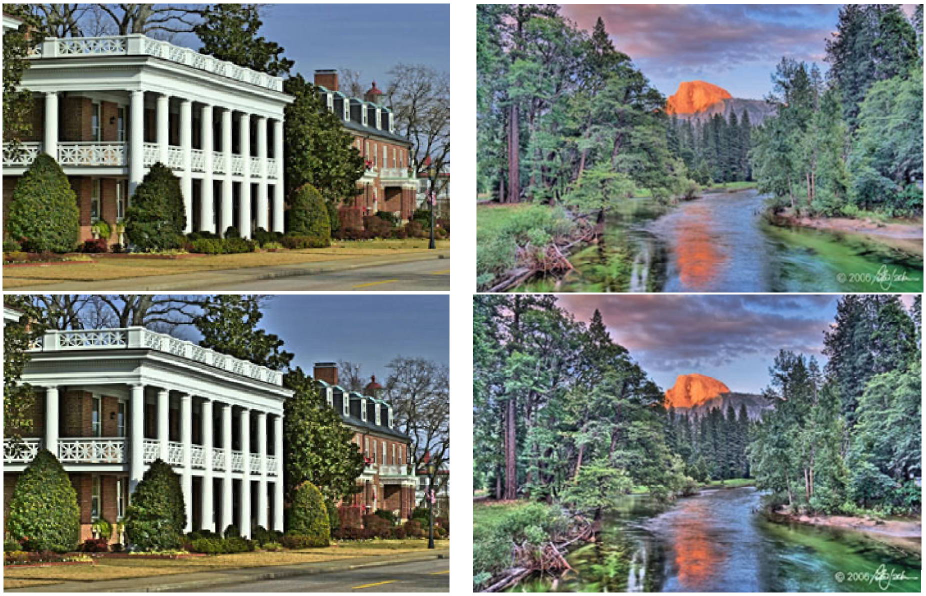


Fig. 4. (*continued*)

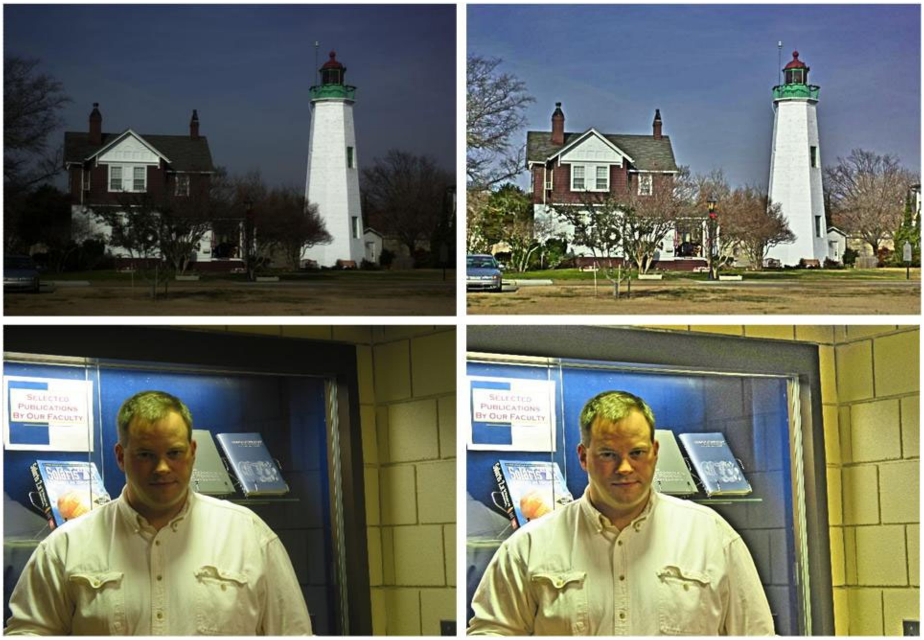

SELECTED PUBLICATIONS BY OUR FACULTY
SELECTED PUBLICATIONS BY OUR FACULTY

Fig. 5. More results of image enhancement using proposed method

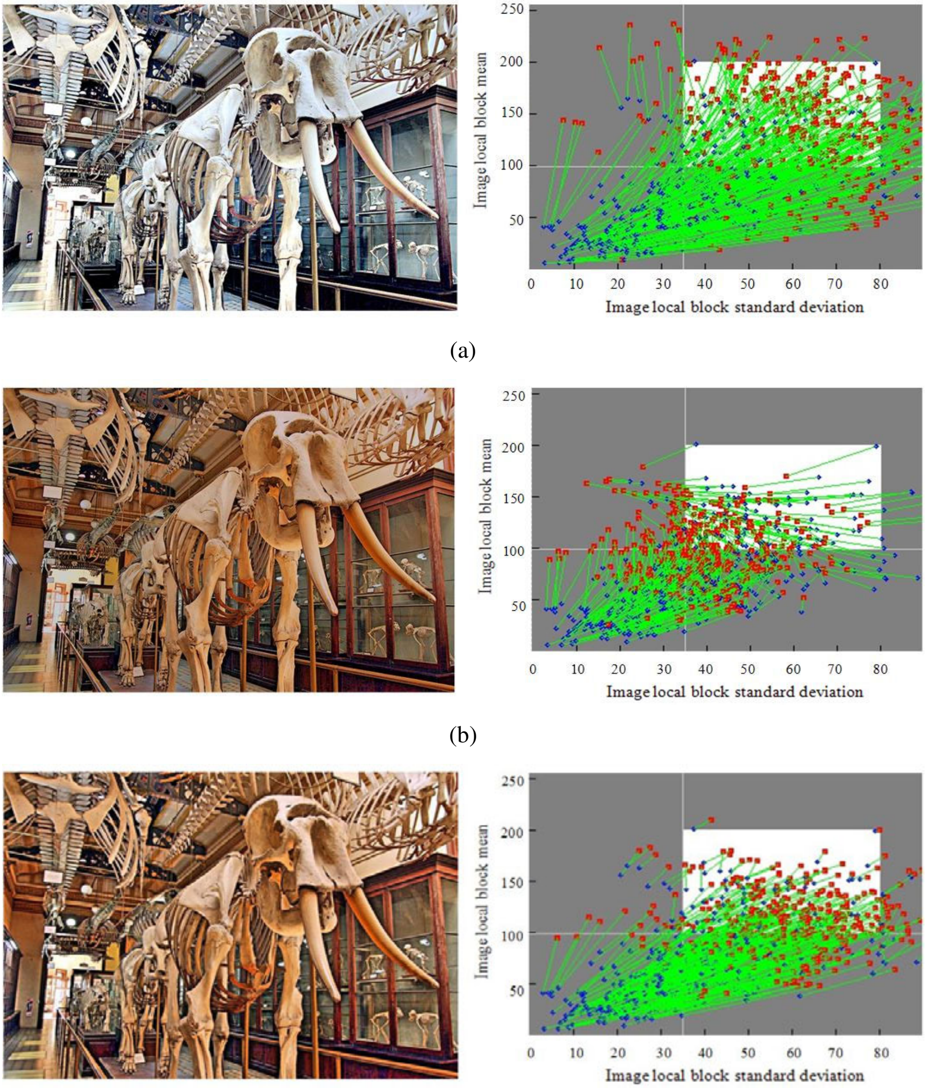
3.2 Objective Evaluation

In this subsection, we use the objective evaluation criteria to compare the performance of the proposed method with other methods. One of the objective evaluation criteria was taken to be the Detail Variance (DV) and Background Variance (BV) from [14]. DV and BV values are obtained firstly by computing the variance of the gray-levels in the neighboring pixels of each pixel in the image. After that, the pixel is classified to the foreground when the variance is more than a threshold; otherwise the pixel is classified to the background. DV is the average variance of the pixels included in the detail region, and BV is the average variance of all the pixels included in the background region. The desired result is increase in DV and no change in BV after applying the enhancement [14]. Here the size of the neighborhood is chosen to be 7 × 7 and threshold was chosen to be 5. Results for some test images are shown in Table 1. From Table 1 it is clear that the results are better than AINDANE and Literature [13], and are comparable to the MSRCR. The variance of the detail region is increased sufficiently and variance of the background region is unchanged, which is desirable. Even the result of MSRCR seems best, from the subjective evaluation it is clear that MSRCR produces unnatural result with unnecessarily increase in contrast.

The proposed image enhancement method is also compared with other methods by using statistical method proposed by D. J. Jabson et al. [15]. In this method, the statistical properties of image, mean, and the mean of zonal standard deviation, are used to describe the visual quality of the image in terms of image contrast and details. Here, first we divide the image into 40 × 40 non overlapping pixel blocks and for each block mean and standard deviation is calculated and plotted as shown in Fig. 6 with different enhancement results. The image quality is classified as visually optimal if it lies inside the white rectangle region [15]. The blue data points indicates the position of the small block of the original image and corresponding red data points connected via a straight line indicates the position of those small blocks of the image after enhancement. Fig. 6 (a) shows the enhanced image using MSRCR and corresponding statistical plot of that image, Fig. 6 (b) shows the enhanced image using AINDANE and corresponding statistical plot and Fig. 6 (c) shows the enhanced image using proposed method and corresponding statistical plot. By using proposed method *50 %* of the red points are inside the white rectangle where as only *32 %* and *47 %* of the red points is inside white rectangle using AINDANE and MSRCR enhancement methods respectively. This proves the robustness of the proposed method in comparison with other methods.

Table 1. DV and BV values for different enhancement results

Image	**Original**		**MSRCR [7]**		**AINDANE [9]**		**Literature [13]**		**Proposed Method**	
	BV	DV	BV	DV	BV	DV	BV	DV	BV	DV
1	1.52	15.03	1.50	35.80	1.09	17.37	1.23	21.10	1.21	28.59
2	2.80	23.95	2.83	41.60	3.39	26.11	3.47	36.38	3.47	36.38
3	3.27	25.18	3.21	57.40	3.54	35.87	3.47	40.78	3.50	54.30
4	2.48	21.56	1.27	38.91	1.82	24.15	1.92	28.09	1.69	39.30
Average	**2.51**	**21.43**	**2.20**	**43.42**	**2.46**	**25.87**	**2.52**	**31.58**	**2.46**	**39.64**



(a)

(b)

(c)

Fig. 6. Enhancement results with statistical characteristics of image of Fig. 3 (a) (image local block mean versus corresponding local block standard deviation) using: (a) MSRCR method, (b) AINDANE method, and (c) proposed method

4 Conclusion

In this paper we propose a color image enhancement method in HSV space with preserving image details. Both subjective and objective performance evaluation has shown the proposed method is superior then other enhancement methods like

MSRCR, AINDANE and [13]. The enhancement results are natural with no change in original color, because the use of HSV space for image enhancement without changing hue and saturation of the input image. The future work will focus on decreasing computational complexicity and making the enhancement results more natural looking without degrading quality of enhancement results.

References

1. Kim, Y.-T.: Contrast Enhancement Using Brightness Preserving Hi-Histogram Equalization. IEEE Trans. on Consumer Electronics 43, 1–8 (1997)
2. Wang, Y., Chen, Q., Zhang, B.: Image Enhancement Based on Equal Area Dualistic Sub-Image Histogram Equalization Method. IEEE Trans. on Consumer Electronics 45, 68–75 (1999)
3. Chen, S.-D., Ramli, R.: Contrast Enhancement using Recursive Mean-Separate Histogram Equalization for Scalable Brightness Preservation. IEEE Trans. on Consumer Electronics 49, 1301–1309 (2003)
4. Wang, Q., Ward, R.K.: Fast Image/Video Contrast Enhancement Based on Weighted Threshold Histogram Equalization. IEEE Trans. on Consumer Electronics 53, 757–764 (2007)
5. Fattal, R., Lischinski, D., Werman, M.: Gradient Domain High Dynamic Range Compression. In: Proc. of the 24th Annual Conference on Computer Graphics and Interactive Technologies, pp. 249–256 (2002)
6. Debevec, P.E., Malik, J.: Recovering High Dynamic Range Radiance Maps from Photographs. In: Proc. of the 24th Annual Conference on Computer Graphics and Interactive Technologies, pp. 369–378 (1997)
7. Jobson, D.J., Rahman, Z., Woodell, G.A.: A Multiscale Retinex for Bridging the Gap between Color Images and the Human Observation of Scenes. IEEE Trans. on Image Processing 6, 965–976 (1997)
8. Choi, D.H., Jang, I.H., Kim, M.H., Kim, N.C.: Color Image Enhancement using Single-Scale Retinex based on an Improved Image Formation Model. In: 16th Ruropean Conf. (EUSIPCO 2008), Lausanne, Switzerland (2008)
9. Tao, L., Asari, V. K.: Adaptive and Integrated Neighborhood-Dependent approach for Nonlinear Enhancement of Color Images. J. of Electronic Imaging 14, 043006-1–043006-14 (2005)
10. Xiao, D., Ohya, J.: Contrast Enhancement of Color Images Based on Wavelet Transform and Human Visual System. In: Proc. of the IASTED Int. Conf. Graphics and Visualization in Engg., Florida, USA, pp. 58–63 (2007)
11. Clement, J.C., Parbukumar, M., Baskar, A.: Color Image Enhancement in Compressed DCT Domain. ICGST-GVIP Journal 10, 31–38 (2010) ISSN: 1687-398X
12. Monobe, Y., Yamashita, H., Kurosawa, T., Kotora, H.: Dynamic Range Compression Preserving Local Image Contrast for Digital Video Camera. IEEE Trans. on Consumer Electronics 51, 1–10 (2005)
13. Ghimire, D., Lee, J.: Nonlinear Transfer Function-Based Local Approach for Color Image Enhancement. IEEE Trans. on Consumer Electronics 57, 858–865 (2011)
14. Ramponi, G., Strobel, N.K., Mitra, S.K., Yu, T.-H.: Nonlinear Unsharp Masking Methods for Image Contrast Enhancement. J. of Electronic Imaging 5, 353–366 (1996)
15. Jobson, D.J., Rahman, Z., Woodell, G.A.: Statistic of Visual Representation. In: Proc. of SPIE, vol. 4736, pp. 25–35 (2002)

3D Perception Adjustment of Stereoscopic Images Based upon Depth Map

Jong In Gil, Seung Eun Jang, and Manbae Kim

Department of Computer and Communications Engineering,
Kangwon National University,
Chunchon, 200-701, Repubic of Korea
{jigil,jse4485,manbae}@kangwon.ac.kr

Abstract. Recently, a variety of stereoscopic contents have been provided to academic and industrial fields for broadcasting, movies and mobile materials. However, few works have been interested in the adjustment of 3D contents for diverse displays. For instance, movie contents suited to large screen frequently do not deliver the same 3D perception to small-size screen such as mobile phone, tabular PCs, etc. For this, this paper presents an adjustment method of stereoscopic contents. 2D+Depth is one of popular methods with which stereoscopic images are generated. For this, depth planes are derived based on a depth histogram. By adjusting depth planes, a new depth map is made. Then 2D+Depth produces a stereoscopic image. Experiments performed on various 2D+Depth images validate that the proposed methods deliver more enhanced 3D depth based on subjective evaluation experiments.

Keywords: stereoscopic perception, depth map adjustment, subjective test.

1 Introduction

The advances in stereoscopic video technologies have led to an increasing interest in various 3D applications [1, 2]. Significant amount of research has been carried out for new 3D applications. In general, stereoscopic images are acquired from two camera sensors. Displaying the images on a 3D monitor, humans can view and perceive 3D. In the previous applications, the stereoscopic images are delivered to viewers without any modification or enhancement. Any similar efforts have not been performed for solving such problem, yet. Based on this, this paper presents a novel method to enhance 3D perception of the stereoscopic images based upon depth map. The overall aim of the proposed method is to enhance the quality of viewing experience of the end users [3]. *2D+D* (depth map) approach is used as the representation format in our approach. Spatial complexity of depth map is one of the key dimensions by which the perceived quality and depth perception of stereoscopic image are adjusted. The experimental results demonstrate that the lower the spatial complexity is, the higher the perceived video quality and depth perception are. In order to support the assertion, human visual fatigue is also examined.

The paper is organized as follows: Overall approach is introduced in Section 2. Section 3 presents the algorithm of dividing the depth into depth planes utilizing the

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 13–21, 2011.

spatial complexity of the depth maps. The depth map adjustment algorithm is presented in Section 4. Experimental results are described in Section 5. Finally, Section 6 concludes the paper.

2 Overview of Proposed Method

Fig. 1 shows the overall approach of the proposed method. Given an input depth map, its histogram is analyzed for separating a depth map into multiple depth planes. The spatial complexity is examined for the depth planes. Then the depth planes undergo the adjustment for the variation of 3D perception. Combining the depth planes, a new depth map is made. Finally, a stereoscopic image is generated by 2D+Depth method.

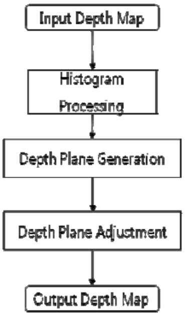


Fig. 1. Block diagram of the proposed method

2.1 Spatial Complexity

Spatial complexity of a depth map is measured by calculating a standard deviation of pixel depth values. The reason behind using the standard deviation for the measurement of spatial complexity is that it is the measure of the dispersion or variability of a set of values around the mean of that set [4]. Thus, if the depth map has high spatial complexity, the standard deviation of the pixel depth values is expected to be high. The pixels in the depth map determine the distance of the associated color image pixel to the viewer. They take grey values ranging from 0 to 255. 0 represents the furthest away pixel from the viewer, while 255 corresponds to the closest pixel to the viewer in a 3D scene.

Given an MxN depth map, the mean pixel depth is computed by

$$\mu_D = \frac{1}{NM}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} D(m,n) \tag{1}$$

Subsequently, the standard deviation is defined by

$$\sigma_D = \sqrt{\frac{1}{NM}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} [D(m,n) - \mu_D]^2} \tag{2}$$

2.2 Histogram Analysis

Depth map histogram $H(i)$ provides the frequency of the depth value *i* in the depth map *D*, and is defined as follows:

$$H(i) = \frac{1}{NM} \sum_{m=0}^{M\ 1} \sum_{n=0}^{N\ 1} \delta[i, D(m,n)] \tag{3}$$

where

$$\delta(a,b) = \begin{cases} 1, \text{if} \ \ \mathrm{a} = \mathrm{b} \\ 0, \text{otherwise} \end{cases}$$

As well, a cumulative histogram *C*(*i*) of a histogram *H*(*i*) is defined as

$$C(i) = \sum_{h=0}^{i} H(h) \tag{4}$$

where C(255) = NM for a 256-level luminance image

2.3 Depth Plane Generation

The histogram separation using Gaussian mixture model has been studied in some applications [5]. This method might work for a couple of objects and background. On the contrary, for images containing multiple objects and background, this method may not work well. Therefore, we use a simple, but efficient method utilizing the cumulative histogram. The following condition is used.

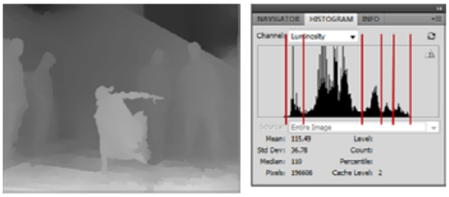
$$| C(i-k) - C(i+k) | < T \tag{5}$$

where *T* is a threshold value and k is a user-defined parameter.

i values satisfying the above condition are chosen as depth thresholds separating a depth map. Suppose that (L+1) depth thresholds (e.g., i_0, i_1, … , i_L) are acquired. L depth planes are then generated. Then the range of the *l*th depth plane is defined as

$$R_D^l = [D_{\min}^l, D_{\max}^l] = [i_{l-1}, i_l], l \in \{1, \ldots, L\} \tag{6}$$

Fig. 2 shows an example of depth thresholds with which a depth map is divided into five depth planes. The red bar indicates the depth thresholds.



Channel: Luminosity

Source: Entire Image

Mean: 115.49 Level:
Std Dev: 36.78 Count:
Median: 110 Percentile:
Pixels: 196608 Cache Level: 2

Fig. 2. The separation of a depth map into depth planes by depth thresholds (colored in red)

The mean and standard deviation of pixel depth values of each depth plane are computed by

$$\mu^l = \frac{\sum_{j \in R_D^l} D(j)}{\text{No.of pixels in } R_D^l} \quad \text{and} \quad \sigma^l = \sqrt{\frac{\sum_{j \in R_D^l} [D(j) - \mu^l]^2}{\text{No.of pixels in } R_D^l}} \tag{7}$$

3 Depth Map Adjustment

The block diagram of Fig. 3 shows the depth map adjustment algorithm proposed in this paper. Given depth planes, standard deviation representing the spatial complexity is computed for each depth plane. The source standard deviation σ_S is the sum of depth plane standard deviations. If a target standard deviation σ_T is determined ($\sigma_T < \sigma_S$), the depth range of depth planes are reduced until σ_S is less than σ_T. As a result, the distance between neighboring depth planes are widened and the 3D depth between them becomes stronger. Finally, a stereoscopic image can be generated from 2D+Depth approach.

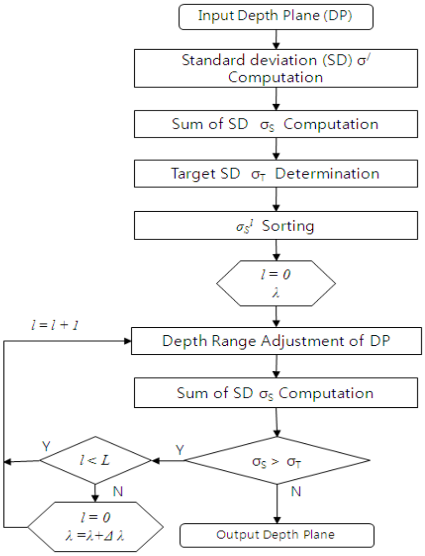

Input Depth Plane (DP)
Standard deviation (SD) σ^l Computation
Sum of SD σ_S Computation
Target SD σ_T Determination
σ_S^l Sorting
l = 0
λ
l = l + 1
Depth Range Adjustment of DP
Sum of SD σ_S Computation
Y
l < L
Y
σ_S > σ_T
N
N
l = 0
λ = λ+Δλ
Output Depth Plane

Fig. 3. The block diagram of depth map adjustment method

For each depth plane, the standard deviation σ^l is computed using Eq. (2). Then σ_S is the sum of L depth plane standard deviations.

$$\sigma_S = \sum_{l=1}^{L} \sigma^l \tag{8}$$

To reduce the spatial complexity, we define a target standard deviation σ_T as follows;

$$\sigma_T = \tau \cdot \sigma_S \tag{9}$$

where τ is a user-defined parameter at [0, 1]. In the experiments, τ is set to be 0.9, 0.8, 0.7, and 0.6.

Until σ_T is achieved, the depth range of depth planes are reduced. The depth planes are sorted according to its standard deviation. The reduction of depth range starts from a depth plane with the greatest standard deviation with a reduction ratio λ. The following equation explains how the range of a depth plane is reduced. For a range [$D^l_{\min}$, $D^l_{\max}$], its depth range is adjusted into [$E^l_{\min}$, $E^l_{\max}$] as follows:

$$E^l_{\min} = (1+\lambda)D^l_{\min} \quad E^l_{\max} = (1-\lambda)D^l_{\max} \tag{10}$$

The depth plane adjustment algorithm is implemented by the following iterative method:

Given *L* input depth planes,

Step 1: σ^l is computed for each depth plane. Subsequently, σ_S is also computed.

Step 2: Target standard deviation σ_T is set with.

Step 3: We sort the depth planes according to σ^l. The depth adjustment of a depth plane with the greatest standard deviation is processed. *l* = 0 and λ is set to be 0.1 or 0.05.

Step 4: The depth range of *l* th depth plane is adjusted. As well, a new standard deviation σ_S is also computed.

Step 5: If $\sigma_T < \sigma_S$, examine whether *l* is less than *L*. If *l* < *L*, *l* = *l* + 1 and go to Step 4. Otherwise *l* is 0 and increase *l* by 1 and $\lambda = \lambda + \Delta\lambda$.

Step 6: If $\sigma_T > \sigma_S$, stop and final depth planes are acquired.

4 Experimental Results

The proposed method was performed on various 2D images and depth maps. We illustrate the results for each test image. The first image is MSR *breakdance* image [6] and depth sequences as shown in Fig. 4.

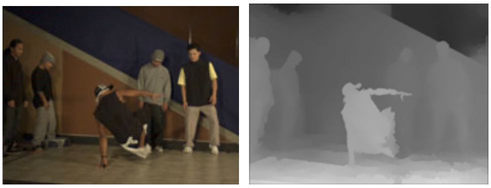


Fig. 4. RGB image and depth map [6]

The depth thresholds are 38, 45, 70, 149, 210, and 216. Five depth planes are shown in Fig. 5.

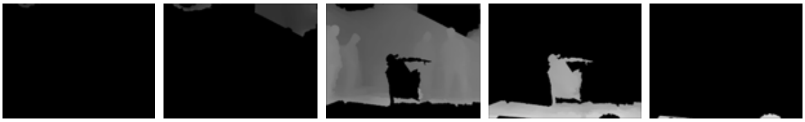
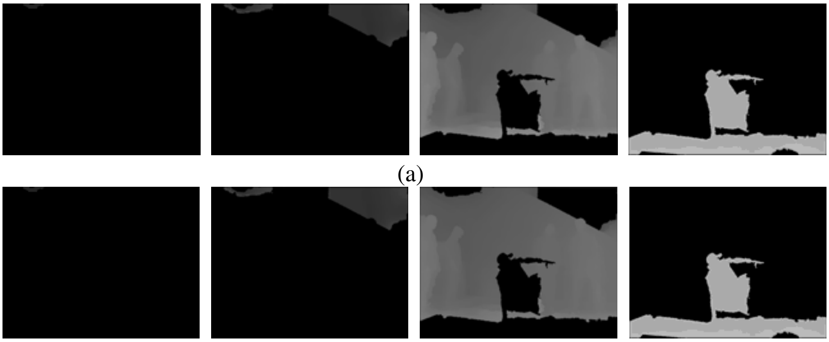


Fig. 5. Five depth planes of Breakdance

Table 1. Standard deviation of input and output depth planes. (* denotes the range-changed depth planes).

		Output standard deviation		
Depth plane	σ^l	$\tau = 0.9$ σ_T =998	$\tau = 0.8$ σ_T =887	$\tau = 0.7$ σ_T =776
1	43	43	39^*	39^*
2	8	82	68^*	68^*
3	35	359	289^*	188^*
4	41	298^*	176^*	176^*
5	214	214	214^*	214^*
$\sigma_S = \Sigma\sigma^l$	341	996	786	685

When τ is 0.8, at the first iteration, we can not achieve σ_T. So, we increased λ by 0.1. Then in the second iteration, the condition was met at the depth plane 4. A final σ_S is 786. For τ =0.7, the second iteration was completed with depth plane 3. A final σ_S is 685. Detailed numerical values are found in Table.1. The final depth planes are shown in Fig. 6.

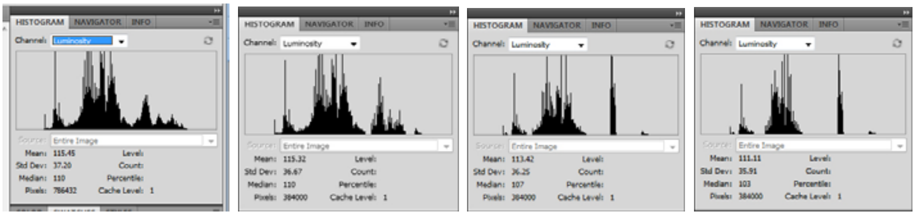


(a)

(b)

Fig. 6. Depth planes generated according to τ (a) τ =0.8 and (b) τ =0.7

Fig. 7 shows the histogram of input and output depth maps. As τ becomes smaller, the distance between depth planes increases. Therefore, depth difference is more apparent.



(a)

Channel: Luminosity
Source: Entire Image
Mean: 115.45 Level:
Std Dev: 37.20 Count:
Median: 110 Percentile:
Pixels: 786432 Cache Level: 1

(b)

Channel: Luminosity
Source: Entire Image
Mean: 115.32 Level:
Std Dev: 36.67 Count:
Median: 110 Percentile:
Pixels: 384000 Cache Level: 1

(c)

Channel: Luminosity
Source: Entire Image
Mean: 113.42 Level:
Std Dev: 36.25 Count:
Median: 107 Percentile:
Pixels: 384000 Cache Level: 1

(d)

Channel: Luminosity
Source: Entire Image
Mean: 111.11 Level:
Std Dev: 35.91 Count:
Median: 103 Percentile:
Pixels: 384000 Cache Level: 1

Fig. 7. The histograms of (a) input depth map and depth maps at (b) τ =0.9, (c) τ =0.8, and (d) τ =0.7

The second test image in Fig. 8 is *Ballet* sequence of MSR [6]. Fig. 9 shows newly adjusted depth planes.

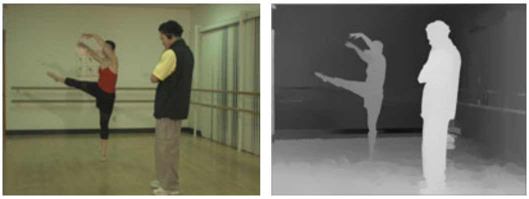
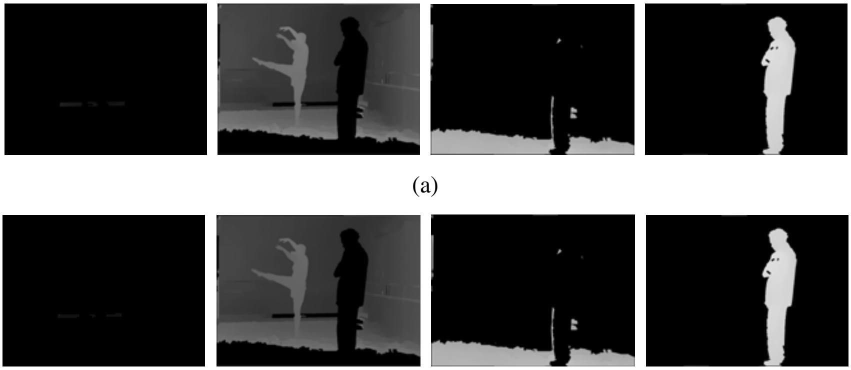


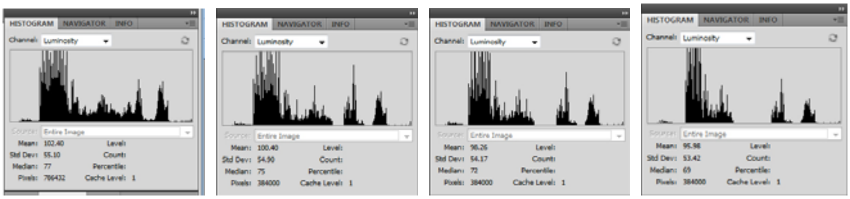
Fig. 8. RGB image and depth map of *Ballet*



(a)

(b)

Fig. 9. Depth planes generated according to τ (a) τ =0.8 and (b) τ =0.7



(a) Mean: 102.40 Std Dev: 55.10 Median: 77 Pixels: 786432 Level: Count: Percentile: Cache Level: 1

(b) Mean: 100.40 Std Dev: 54.90 Median: 75 Pixels: 384000 Level: Count: Percentile: Cache Level: 1

(c) Mean: 98.26 Std Dev: 54.17 Median: 72 Pixels: 384000 Level: Count: Percentile: Cache Level: 1

(d) Mean: 95.98 Std Dev: 53.42 Median: 69 Pixels: 384000 Level: Count: Percentile: Cache Level: 1

Fig. 10. The histograms of (a) input depth map and depth maps at τ = (b) 0.9, (c) 0.8, and (d) 0.7

We observed the stereoscopic images with a 3D monitor adopting DQCQS (Double Stimulus Continuous Quality Scale) subjective test [7]. At the first stage, original views were displayed to five participants. Each participant watched the views for 10 seconds and their new views for the same period, and evaluated the effect of the 3D depth. Two test sets were carried out in order to examine the 3D perception improvement. Depth perception was then subjectively judged on a scale of 1 (no improvement), 2 (mild improvement), 3 (average improvement), 4 (good improvement) and 5 (excellent improvement) in terms of 3D perception. Fig. 11 shows two subjective grades with respect to τ as well as σ_S. As τ decreases, we observe that the perceived quality is improved. Furthermore, in order to examine the visual fatigue of the stereoscopic image viewing, we performed subjective visual

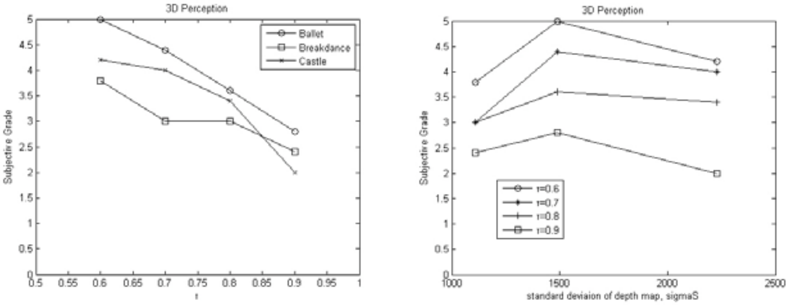


Fig. 11. Subjective grades with respect to (a) τ and (b) σ_S

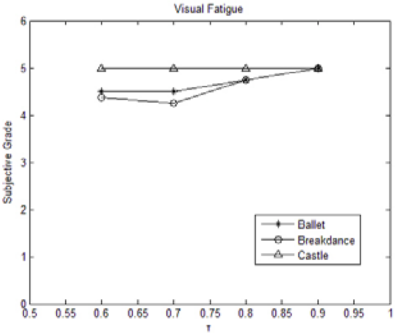


Fig. 12. Subjective test for visual fatigue with respect to τ

fatigue test. Visual fatigue was subjectively judged on a scale of 1 (severe fatigue), 2 (fatigue), 3 (mild fatigue), 4 (slight fatigue) and 5 (not at all). As validated in Fig. 12, the overall grade is greater than 4.0, which means that the proposed method makes comfortable stereoscopic images.

5 Conclusion

In this paper, we presented a depth map adjustment method that could provide the improvement of 3D stereoscopic perception. For this, a histogram of a depth map is used for the extraction of multiple depth planes. For spatial complexity, standard deviation of each depth plane is examined. According to the target standard deviation, the depth range of each depth plane is adjusted, thereby making the distance between neighboring depth planes increased. This effect delivers better 3D perception that was validated through subjective tests. Our proposed method is nearly automatic and is expected to provide a technical contribution to 3D video field.

Acknowledgments. This research was supported by the Ministry of Education, Science Technology (MEST) and National Research Fund (NRF) through the Human Resource Training Project for Regional Innovation and by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency). (NIPA-2011-(C1090-1111-0003))

References

1. Meesters, IJsselsteijn, W.A., Seuntiens, P.J.H.: A survey of perceptual evaluations and requirements of three-dimensional TV. IEEE Transactions on Circuits and Systems for Video Technology 14(3), 381–391 (2004)
2. Blonde, L., Doyen, D., Borel, T.: 3D stereo rendering challenges and techniques. In: IEEE 2010 44th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6 (2010)
3. Fehn, C.: Depth-Image-Based Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV. In: Proc. of the SPIE, CA, U.S.A, vol. 5291, pp. 93–104 (January 2004)
4. Devore, F.: Probability and Statistics for Engineering and the Sciences, Duxbury (1995)
5. Harimi, A., Ahmadyfard, A.: Image Segmentation Using Correlative Histogram Modeled by Gaussian Mixture. In: IEEE Int' Conf. on Image Processing (2009)
6. `http://research.microsoft.com/vision/InteractiveVisualMediaGroup/3DVideoDownload/`, Microsoft Research
7. Lee, E., Heo, H., Park, K.: The comparative measurements of eyestrain caused by 2D and 3D displays. IEEE Trans. on Consumer Electronics 56(3), 1677–1683 (2010)

Super-Resolved Free-Viewpoint Image Synthesis Using Semi-global Depth Estimation and Depth-Reliability-Based Regularization

Keita Takahashi[1] and Takeshi Naemura[2]

[1] The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan
keita.takahashi@ieee.org
http://nae-lab.org/~keita/
[2] The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Abstract. A method for synthesizing high-quality free-viewpoint images from a set of multi-view images is presented. First, an accurate depth map is estimated from a given target viewpoint using modified semi-global stereo matching. Then, a high-resolution image from that viewpoint is obtained through super-resolution reconstruction. The depth estimation results from the first step are used for the second step. First, the depth values are used to associate pixels between the input images and the latent high-resolution image. Second, the pixel-wise reliabilities of the depth information are used for regularization to adaptively control the strength of the super-resolution reconstruction. Experimental results using real images showed the effectiveness of our method.

Keywords: free-viewpoint image, semi-global stereo, super-resolution, depth reliability, regularization.

1 Introduction

Free-viewpoint image synthesis refers to the process of combining a set of multi-view images to generate an image from a new viewpoint where no camera was actually located. This technology has attracted much research interest due to its potential for representing 3-D visual information [1]; using this technology, users can fly through the 3-D space and can also display real objects on autostereoscopic 3-D displays with tens of parallax views [2].

In this work, we reconsider the framework of free-viewpoint image synthesis. In general, this synthesis consists of two steps: first, the depth/shape of the target scene is estimated from input images; then, using the estimated depth/shape, the input images are registered and blended together to produce a new image. The blending operation in the second step can obscure depth/shape errors by blurring the image. However, this scheme has a fundamental limitation in the resolution of the resultant image; fine textures are decayed due to the blurring nature of blending.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 22–35, 2011.

A promising solution for improving resolution is to replace blending by super-resolution (SR) reconstruction [3] because multiple observations of the same scene are given as input. However, SR reconstruction is very sensitive to registration errors; it could even be destructive if applied with large registration errors. Estimating perfect depth/shape information from images alone is far beyond the capability of current computer vision technologies, so some extent of registration errors should be accepted. Consequently, we conceived of the idea to combine blend-based synthesis and SR-based synthesis adaptively according to the reliability of the estimated depth information.

On the basis of this idea, we propose a new method for super-resolved free-viewpoint image synthesis. Our method has three features. First, we adopt a view-dependent approach like that in Refs. [4,5]; we focus on image synthesis from the given target viewpoint rather than complete reconstruction of the 3-D structure. More precisely, our method works directly on the coordinate system of the target free-viewpoint image. The second feature is semi-global depth estimation based on that in Refs. [6,7], which achieves accurate depth estimation with considerably low computational costs. The final feature is depth-reliability-based regularization, which can control the strength of SR reconstruction according to the pixel-wise reliability of the depth information. This regularization is the key to achieving high-quality synthesis and also provides a new framework where blend-based synthesis and SR-based synthesis are adaptively combined. The effectiveness of our method was confirmed by experiments using real images.

1.1 Background

Super-resolution (SR) reconstruction [3] combines multiple low-resolution images to restore a latent high-resolution image. One of the input images is selected as the basis image to which other input images are registered and for which the resulting high-resolution image is synthesized. Then, an image formation model is established between the input and latent high-resolution images. Finally, by inverting the image formation model with prior knowledge, the latent high-resolution image can be restored. However, the viewpoint of the resulting image is limited to that of one of the input images because this technology is not designed for producing free-viewpoint images.

Free-viewpoint image synthesis has been studied in a different context [1]. As mentioned above, most conventional methods use blend-based synthesis, resulting in the fundamental limitation of the image resolution. To our knowledge, only a few works use SR reconstruction for free-viewpoint image synthesis. Tung et al. [8] super-resolved input multi-view images, and Goldluecke et al. [9] synthesized texture maps using SR reconstruction. Their purpose was to generate a complete 3-D model of a single object. In contrast, our method takes a view-dependent approach for synthesizing free-viewpoint images and deals with the entire scene (which includes both objects and backgrounds). The most similar work to ours is that of Mudenagudi et al. [10]. They formulated view-dependent SR reconstruction of an entire scene as a multiple-labeling problem, where a label corresponds to the color of each pixel of the resulting image. However,

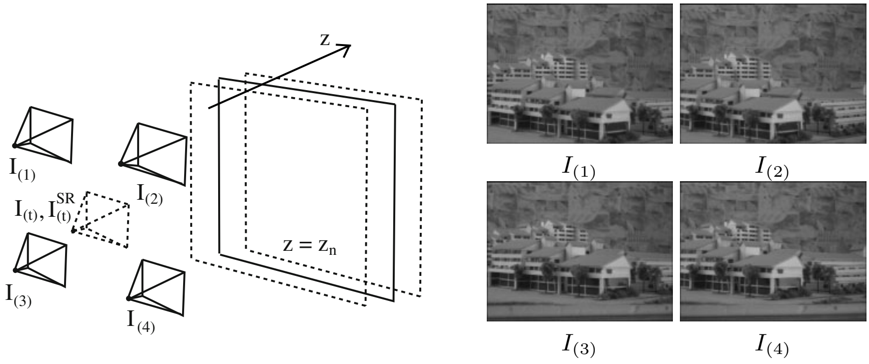


Fig. 1. Configuration used in proposed method (left) and input images (right)

their method was computationally complex and expensive due to the nature of the formulation. Our method, which consists of view-dependent depth estimation followed by SR reconstruction with depth-reliability-based regularization, is computationally more tractable and would be easier to speed-up for real-time applications in the future. Our previous work in Ref. [11] also aimed for view-dependent SR reconstruction. But due to the poor depth estimation and non-efficient algorithm design, it is incomparable to the method presented in this paper.

2 Overview of Proposed Method

The configuration used by our method is shown in Fig. 1. The input images, denoted by $I_{(m)}$ $(m = 1, \dots M)$, are captured from viewpoints that are arranged roughly on the same plane. The camera parameters are estimated beforehand. The distance from the input camera plane is denoted by z. The goal of our method is to synthesize an image viewed from a new viewpoint, referred to as the target viewpoint, which is denoted by t. We define two synthesized images, $I_{(t)}$ and $I^{SR}_{(t)}$. $I_{(t)}$ is produced by blend-based synthesis and has the same resolution as the input images; $I^{SR}_{(t)}$ is produced by our new SR-based scheme. We assume that four images are given as the input and that the resolution of $I^{SR}_{(t)}$ is twice that of $I_{(t)}$ both in the horizontal and vertical directions. However, our framework can easily be extended to more general setups.

In general, our method first registers the input images to the coordinate system of the target viewpoint t and then applies a SR scheme to obtain a high-resolution resulting image. Registration of multi-view images is equivalent to depth estimation. In particular, if pixel-wise depth information from the target viewpoint is available, all pixels of the target image can be associated with the pixels of the input images, which is sufficient for constructing an image formation model for SR reconstruction. Thus, the first step of our method is to estimate a depth map viewed from the target viewpoint (described in Section

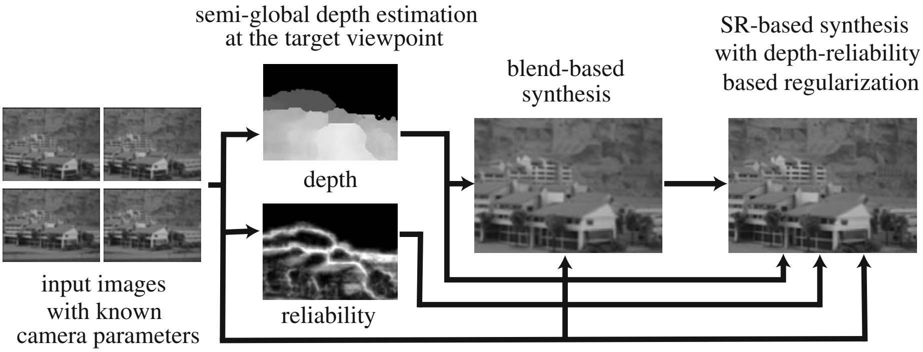

semi-global depth estimation at the target viewpoint
SR-based synthesis with depth-reliability based regularization
blend-based synthesis
depth
reliability
input images with known camera parameters

Fig. 2. Flowchart of our method

3). To estimate accurate depth with reasonable speed, we use the semi-global stereo method [6,7], modified for our problem. The depth map is estimated in the same resolution as the input images. It is then upsampled to the resolution of $I_{(t)}^{SR}$ and used for the next SR reconstruction step (described in Section 4). In SR reconstruction, the per-pixel reliability of the depth information, which is obtained through the depth-estimation step, is also used to control the strength of the regularization. This scheme, referred to as depth-reliability-based regularization, is the key to achieving high-quality synthesis, since it can adaptively combine blend-based synthesis and SR-based synthesis. Our entire method is summarized in Fig. 2.

3 Semi-global Depth Estimation

The first step of our method is to estimate a depth map from the target viewpoint. In Section 3.1, we briefly describe the semi-global stereo method [6,7], which is extended to our free-viewpoint setup in Section 3.2. The obtained depth map is further refined in Section 3.3.

3.1 Semi-global Stereo Matching

The purpose of stereo matching, given two or more input images, is to find pixel correspondences between the images. This is equivalent to depth estimation if the camera parameters are known. Typically, one of the input images is selected as the basis image for which the depth value of each pixel is estimated.

Modern stereo methods consider not only the photometric consistency between the input images (i.e., corresponding pixels should exhibit similar intensities/colors) but also the inter-pixel relations in the estimated depth map (i.e., depth values should not vary drastically except around the object boundaries). These conditions are represented as an energy minimization problem, whose optimal solution can be found using sophisticated numerical methods. The most common choices for optimization are belief-propagation and graph-cut, but they are computationally prohibitively expensive since many iterations are required for convergence [12]. In contrast, semi-global stereo matching [6,7] can find a near-optimal solution with much lower computational cost because no iterative calculations are needed.

The energy function for semi-global stereo matching is described as

$$E_{sm}(D) = \sum_{\boldsymbol{p}} \left\{ C(\boldsymbol{p}, D(\boldsymbol{p})) + \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}},\ |D(\boldsymbol{p}) - D(\boldsymbol{q})| = 1} \lambda_1 + \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}},\ |D(\boldsymbol{p}) - D(\boldsymbol{q})| > 1} \lambda_2 \right\}, \quad (1)$$

where D is the resulting depth map and $D(\boldsymbol{p})$ is the depth of a pixel $\boldsymbol{p}$ that is represented as an integer disparity value. The first term evaluates the photometric consistency between the input images for each pixel $\boldsymbol{p}$ with the assumed depth $D(\boldsymbol{p})$. The second and third terms penalize depth discontinuities; $N_{\boldsymbol{p}}$ is the neighbor of $\boldsymbol{p}$, and λ_1 and λ_2 are non-negative weights, where $\lambda_1 \leq \lambda_2$.

The optimization procedure is very similar to dynamic programming. First, the photometric consistency cost $C(\boldsymbol{p}, n)$ is obtained for all pixels and all depth levels. Then, it is accumulated along the 1-D path with a direction r as

$$\begin{aligned} L_{\boldsymbol{r}}(\boldsymbol{p}, n) = C(\boldsymbol{p}, n) &- \min_k L_{\boldsymbol{r}}(\boldsymbol{p} - \boldsymbol{r},\ k) \\ &+ \min \{ L_{\boldsymbol{r}}(\boldsymbol{p} - \boldsymbol{r}, n),\ L_{\boldsymbol{r}}(\boldsymbol{p} - \boldsymbol{r},\ n-1) + \lambda_1, L_{\boldsymbol{r}}(\boldsymbol{p} - \boldsymbol{r},\ n+1) + \lambda_1, \\ &\qquad \min_k L_{\boldsymbol{r}}(\boldsymbol{p} - \boldsymbol{r},\ k) + \lambda_2 \}. \end{aligned} \quad (2)$$

The accumulated costs for 8 or 16 directions (8 are used in this work) are added to yield $S(\boldsymbol{p}, n)$. Finally, a semi-optimal depth map $D(\boldsymbol{p})$ is obtained through a minimum search over the depth levels for each pixel $\boldsymbol{p}$.

$$D(\boldsymbol{p}) = \arg\min_n S(\boldsymbol{p}, n), \text{ where } S(\boldsymbol{p}, n) = \sum_{\boldsymbol{r}} L_{\boldsymbol{r}}(\boldsymbol{p}, n). \quad (3)$$

In the post-processing step, isolated noises are removed from the resulting depth map, but this step is beyond the scope of this paper.

3.2 Extension to Arbitrary Viewpoint Setups

The semi-global stereo method [6,7] was designed to work on the coordinate system of the basis image that is selected from the input images, similar to most stereo methods. However, our purpose is to estimate a depth map directly from the arbitrary target viewpoint where free-viewpoint image synthesis is performed. In this subsection, the coordinate system of the resulting depth map D is set to the target viewpoint, and we introduce three modifications to the original semi-global stereo method [6,7].

First, disparities cannot uniquely be defined to represent depth in our problem because the target viewpoint is set to an arbitrary position. Instead of using disparities, we quantize the depth space into N levels as

$$\frac{1}{z_n} = \frac{1}{z_{\max}} + \frac{n - 1/2}{N} \left(\frac{1}{z_{\min}} - \frac{1}{z_{\max}} \right) \ (n = 1, \ldots, N), \quad (4)$$

where $z_{\min}$ and $z_{\max}$ are the minimum and maximum of the object depths. This quantization is natural because the disparity space (which is proportional to the inverse of the depth) is evenly divided, similar to most stereo methods. In our method, each pixel of the depth map $D(\boldsymbol{p})$ takes an integer that represents the depth index. The physical depth value for $D(\boldsymbol{p})$ can be written as $z_{D(\boldsymbol{p})}$.

Second, the photometric consistency in Eq. (1) should be given over the coordinate system of the target viewpoint. Consequently, we have to map the pixels from the target viewpoint to the input viewpoints in evaluating the consistencies. Specifically, we define $C(\boldsymbol{p},\, D(\boldsymbol{p}))$ as

$$C(\boldsymbol{p},\, D(\boldsymbol{p})) = \frac{1}{\mathcal{Z}} \sum_{\boldsymbol{q} \in B_{\boldsymbol{p}}} \left\{ \sum_{m \neq m'} C_{m,m'}(\boldsymbol{q},\, D(\boldsymbol{q})) \right\}, \tag{5}$$

where $B_{\boldsymbol{p}}$ is a window centered at $\boldsymbol{p}$, m and m' are the indices of input images, and $\mathcal{Z}$ is a constant for normalization. $C_{m,m'}(\boldsymbol{p},\, D(\boldsymbol{p}))$ evaluates the consistency for a pixel $\boldsymbol{p}$ of the target image as

$$C_{m,m'}(\boldsymbol{p},\, D(\boldsymbol{p})) = \mathrm{diff}\left(I_{(m)}(\mathrm{P}_{t \to m}(\boldsymbol{p}, z_{D(\boldsymbol{p})})),\ I_{(m')}(\mathrm{P}_{t \to m'}(\boldsymbol{p}, z_{D(\boldsymbol{p})}))\right), \tag{6}$$

where $P_{\alpha \to \beta}(\boldsymbol{p}, z)$ is a function that maps a point $\boldsymbol{p}$ on the camera α onto the camera β with a known depth z. The derivation details are described in the appendix A.1. The function diff is defined as

$$\mathrm{diff}(a, b) = \min\left\{\|a - b\|^2, \mathrm{diff}_{\max}\right\}, \tag{7}$$

where $\mathrm{diff}_{\max}$ is an upper limit for the difference values. Giving an upper limit is useful for handling occlusions because we have multiple pairs of input images.

Third, λ_2 is set to a constant in our method, while it was set to be proportional to the inverse of the image gradient in the original [6,7]. In our problem, the image gradients are unavailable directly because the coordinate system is set to a new viewpoint from which no image was captured.

3.3 Depth Refinement

The depth map D obtained through the previous step takes discrete integer values. These values can be refined by fitting parabolic curves to the energy values $S(\boldsymbol{p}, n)$ around the minimums. For each pixel $\boldsymbol{p}$, the refined depth value $\hat{D}(\boldsymbol{p})$ and corresponding energy value $\hat{S}_{\min}(\boldsymbol{p})$ are given by

$$\hat{D}(\boldsymbol{p}) = D(\boldsymbol{p}) + \frac{S_{\min}^{pre}(\boldsymbol{p}) - S_{\min}^{next}(\boldsymbol{p})}{2(S_{\min}^{pre}(\boldsymbol{p}) - 2S_{\min}(\boldsymbol{p}) + S_{\min}^{next}(\boldsymbol{p}))} \tag{8}$$

$$\hat{S}_{\min}(\boldsymbol{p}) = S_{\min}(\boldsymbol{p}) - \frac{(S_{\min}^{pre}(\boldsymbol{p}) - S_{\min}^{next}(\boldsymbol{p}))^2}{8(S_{\min}^{pre}(\boldsymbol{p}) - 2S_{\min}(\boldsymbol{p}) + S_{\min}^{next}(\boldsymbol{p}))}, \tag{9}$$

where $S_{\min}(p) = S(\boldsymbol{p}, D(\boldsymbol{p}))$, $S_{\min}^{next}(p) = S(\boldsymbol{p}, D(\boldsymbol{p}) + 1)$, and $S_{\min}^{pre}(\boldsymbol{p}) = S(\boldsymbol{p}, D(\boldsymbol{p}) - 1)$ (see appendix A.2 for the derivation). This refinement is equivalent to interpolation in the disparity space because the value of $D(\boldsymbol{p})$ is proportional to the inverse of the depth. The resulting depth map $\hat{D}$ takes continuous values, but the corresponding physical depths can also be obtained from Eq. (4) by simply treating n as a continuous value. Thus, without any inconsistency, the physical depth for $\hat{D}(\boldsymbol{p})$ can be described as $z_{\hat{D}(\boldsymbol{p})}$.

Using the refined depth map $\hat{D}(\boldsymbol{p})$, we can obtain the image from the target viewpoint, $I_{(t)}$, whose resolution is the same as those of the input images.

$$I_{(t)}(\boldsymbol{p}) = \frac{1}{M} \sum_{m} I_{(m)}(P_{t \to m}(\boldsymbol{p}, z_{\hat{D}(\boldsymbol{p})})) \tag{10}$$

This image is referred to as a *blend-based* image because the input images are blended together to produce it.

4 Super-Resolved Free-Viewpoint Image Synthesis

After the depth estimation from the target viewpoint, which was described in Section 3, we have a depth map $\hat{D}$, cost map $\hat{S}_{\min}$, and synthesized image $I(t)$, with the same resolutions as those of the input images. As the pre-process of super-resolution, the images are upsampled to the target resolution by using a standard interpolation method (in this work, bicubic interpolation), to obtain $\hat{D}_{\uparrow}$, $\hat{S}_{\min\uparrow}$, and $I(t)_{\uparrow}$. The super-resolved image from the target viewpoint is denoted as $I^{SR}_{(t)}$; the inference process is described in this section.

4.1 Formulation with Depth-Reliability-Based Regularization

Following the standard reconstruction-based SR scheme, the problem can be described by minimization of a energy function E_{sr} as

$$E_{sr}(I^{SR}_{(t)}) = E^{(1)}_{sr}(I_{(1)}, ..., I_{(M)} | I^{SR}_{(t)}) + \lambda \, E^{(2)}_{sr}(I^{SR}_{(t)}), \tag{11}$$

where $E^{(1)}_{sr}$ is a fidelity term, $E^{(2)}_{sr}$ is a regularizer, and λ is a positive weight.

The fidelity term evaluates the relation between the input images $I_{(m)}$ and the desired super-resolved image $I^{SR}_{(t)}$. We formulated it as

$$\mathrm{E}^{(1)}_{sr} = \sum_{m} \sum_{\boldsymbol{p} \in I_{(m)}} \| I_{(m)}(\boldsymbol{p}) - \hat{I}_{(m)}(\boldsymbol{p}) \|^2, \text{ where } \hat{I}_{(m)} = f_{t_{\uparrow} \to m}(I^{SR}_{(t)}, \ \hat{D}_{\uparrow}). \tag{12}$$

In brief, the function $f_{t_{\uparrow} \to m}$ represents an image formation model. Using the given depth map $\hat{D}_{\uparrow}$, $f_{t_{\uparrow} \to m}$ transforms the latent image $I^{SR}_{(t)}$ into the m-th input image. The pixel correspondences between the two cameras, $t_{\uparrow}$ and m, are captured by $P_{t_{\uparrow} \to m}$, where $t_{\uparrow}$ means the target image has double the resolution. Occlusions and the point-spreading function are also considered in this transform (see appendix A.3 for more details).

The regularizer should reflect the prior knowledge about the resulting image $I^{SR}_{(t)}$, where we introduce two assumptions. First, $I^{SR}_{(t)}$ resembles the upsampled version of the blend-based image $I_{(t)\uparrow}$. Second, the image formation model is less reliable where depth estimation is less accurate. On the basis of these assumptions, we define the regularization term as

$$E_{sr}^{(2)} = \sum_{\boldsymbol{p} \in I_{(t)}^{SR}} w(\boldsymbol{p}) \| I_{(t)}^{SR}(\boldsymbol{p}) - I_{(t)\uparrow}(\boldsymbol{p}) \|^2 \tag{13}$$

$$\text{where } w(\boldsymbol{p}) = \max\{ \| \hat{S}_{\min\uparrow}(\boldsymbol{p}) \|^4, w_{\min} \}. \tag{14}$$

Note that the second assumption is reflected in the pixel-wise weighting factor $w(\boldsymbol{p})$, which introduces adaptivity to the regularization. We observe that $\hat{S}_{\min}(\boldsymbol{p})$ takes large values around occlusion boundaries, for example, where the estimated depths are likely to be erroneous (see Fig. 4 (b)). Thus, for such regions, we increase the weight for the regularization term to stabilize the result. When the weight is ultimately large for a pixel $\boldsymbol{p}$, the result for $\boldsymbol{p}$ converges to the blend-based synthesis, i.e., $I_{(t)}^{SR}(\boldsymbol{p}) \sim I_{(t)\uparrow}(\boldsymbol{p})$. Meanwhile, for the regions where the depth estimation is sufficiently reliable, we decrease the weight for regularization to encourage the resolution enhancement that is enabled by the image formation model. This scheme, referred to as depth-reliability-based regularization, is very important in practice because depth information cannot be perfect. Moreover, this regularization is a natural extension of the conventional blend-based approach since it can adaptively combine blend-based synthesis with SR-based synthesis.

4.2 Implementation

Let X, $\bar{X}$, and Y_m be 1-D vector representations of $I_{(t)}^{SR}$, $I_{(t)\uparrow}$, and $I_{(m)}$, respectively. Let A_m be a matrix that represents the relation between the inputs and outputs of the function $f_{t_\uparrow \to m}$ in Eq. (12). Let W denote a diagonal matrix given by diag(w). Equation (11) can be rewritten as

$$E_{sr}(X) = \sum_m \| Y_m - A_m X \|^2 + \lambda \, W \| (X - \bar{X}) \|^2. \tag{15}$$

We set the initial value of X as $X_0 = \bar{X}$ and iterate

$$X_{j+1} = X_j - \alpha_j \nabla E_{sr}(X_j), \;\; \alpha_j = \frac{\| \nabla E_{sr}(X_j) \|^2}{\nabla E_{sr}(X_j)^T (\nabla^2 E_{sr}) \nabla E_{sr}(X_j)} \tag{16}$$

until it converges, where $\nabla E_{sr}(X_j)$ denotes the gradient of E_{sr} at $X = X_j$. In our test, this solution is faster and more stable than solving a linear equation $\nabla E_{sr}(X_j) = 0$ using MATLAB's numerical solver.

5 Experiments

The four images of a city diorama shown in Fig. 1, which were taken from the *Multi-view Image Database of University of Tsukuba, Japan*, were used as input

Table 1. Default values for parameters

Eq. (1)	$\lambda_1 = 100$, $\lambda_2 = 400$
Eq. (4)	$z_{\min}$=250 mm (21.00*), $z_{\max}$=1900 mm (2.76*), $N = 40$
Eq. (5)	size of $B_{\boldsymbol{p}}$: 3×3 pixels
Eq. (7)	$\mathrm{diff}_{\max} = 150$
Eq. (11)	$\lambda = 5.0\times10^{-13}$
Eq. (14)	$w_{\min} = 10$

∗corresponding disparities (in pixels) between input images

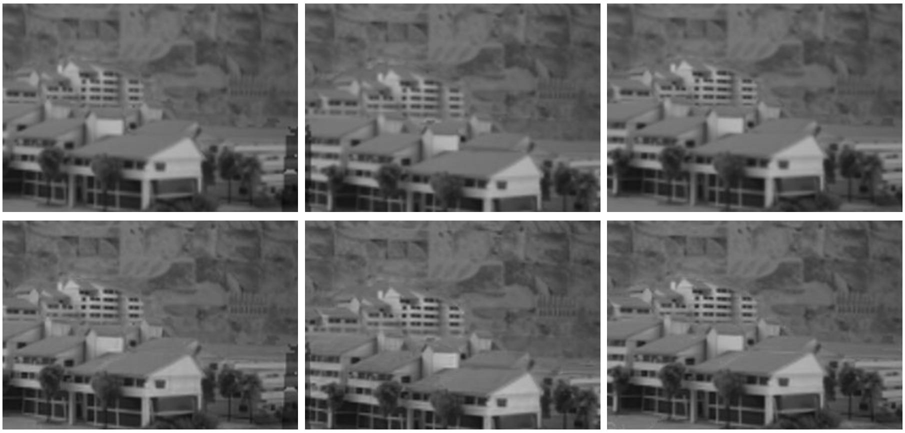
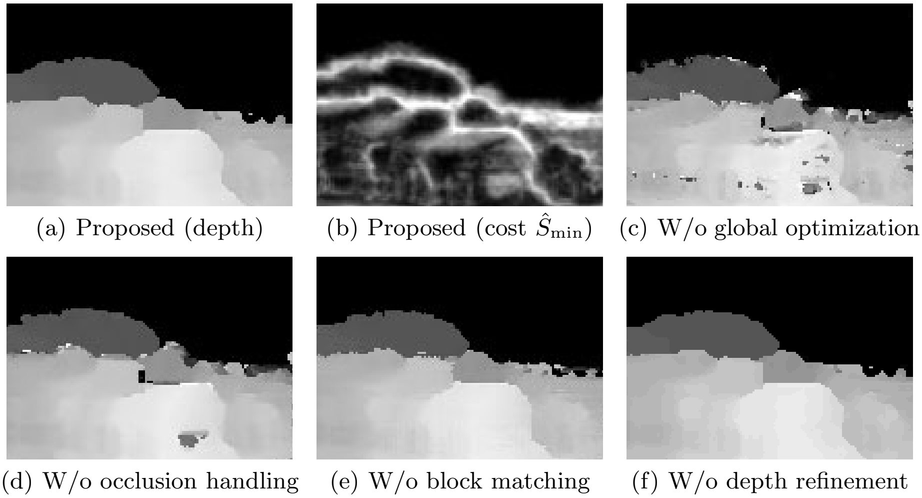


Fig. 3. Resulting images from various viewpoints by (top) blend-based synthesis and (bottom) SR-based synthesis. A demo video is available from our website.

for our method. The input viewpoints were located at the corners of a square of 16 × 16 mm. The original images had 640 × 480 pixels in RGB color. We converted them to grayscale and reduced them to 160 × 120 pixels to use them as the input.

The target viewpoints were located inside the square formed by the input viewpoints. Our method first estimated a depth map with 160 × 120 pixels from a given target viewpoint, then generated a resulting image with 320 × 240 pixels from that viewpoint. The parameter settings, which were empirically determined based on several tests, are in Table 1.

Images from different viewpoints were generated by blend-based synthesis ($I_{(t)\uparrow}$) and SR-based synthesis ($I_{(t)}^{SR}$), shown in the top and bottom rows respectively of Fig. 3. The tuples of numbers below the images indicate the coordinates of the viewpoints according to the database notation, where the input viewpoints were described as (1, 6), (3, 6), (1, 8), and (3, 8). The figure shows that free-viewpoint images were successfully synthesized and that SR-based synthesis achieves better quality with finer texture details. A video for further demonstration is available from our website http://nae-lab.org/˜keita/.


(a) Proposed (depth) (b) Proposed (cost $\hat{S}_{\min}$) (c) W/o global optimization

(d) W/o occlusion handling (e) W/o block matching (f) W/o depth refinement

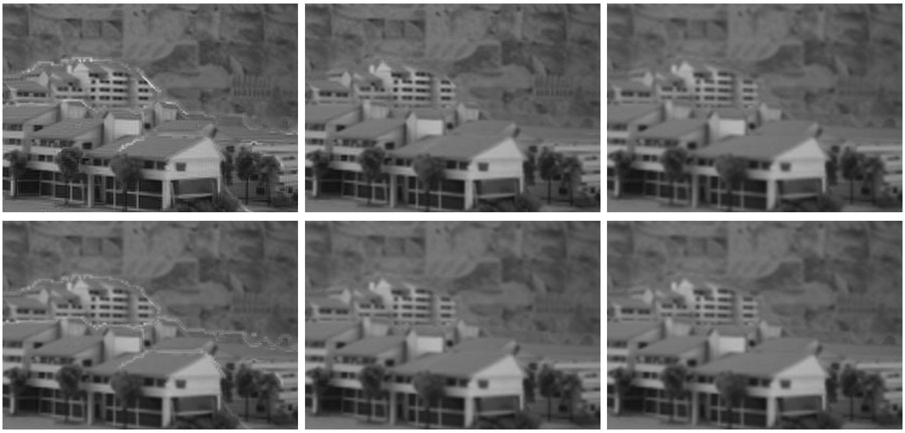
Fig. 4. Comparison of depth estimation results

5.1 Detailed Evaluation

To evaluate our method more closely, we fixed the target viewpoint to the center of the square, i.e., $(2, 7)$ according to the database notation, where the ground truth image was available from the database.

First, we evaluated the depth estimation part of our method. We disabled each element of our method one by one and estimated the depth. The results are shown in Fig. 4. As shown in (a), the proposed method produced a good result. The cost map $\hat{S}_{\min}$, shown in 1/10 scale in (b), was used for the depth-reliability-based regularization mentioned later. When the global optimization was turned off by setting $\lambda_1, \lambda_2 = 0$, the resulting depth map was very noisy, as shown in (c). Unless the occlusions were handled properly, depth estimation was erroneous around the occlusion boundaries, as shown in (d). When the block matching was disabled by setting the block size to 1×1 pixels, the depth map became granular, as shown in (e). When depth refinement was skipped, the depth map took only the quantized values, as shown in (f).

Next, we evaluated the adaptive regularization scheme in SR-based synthesis, which is represented by Eq. (14). Images synthesized with different regularization factors (λ in Eq. (11)) are shown in Fig. 5. The top row shows the results with adaptive regularization, and the bottom shows those without it, where $w(\boldsymbol{p})$ was fixed to 2000 for all pixels. When λ became larger (meaning stronger regularization), the resulting images by SR-based synthesis converged to $I_{(t)\uparrow}$ in both cases. Meanwhile, when λ became smaller (meaning weaker regularization), the resulting images were sharper, but some regions, such as occlusion boundaries, became noisy due to mis-registrations. By our regularization scheme, the resulting quality was successfully optimized around $\lambda = 5.0 \times 10^{-13}$ because the regions with less reliable depth are more strongly regularized. Without this



$\lambda = 5.0 \times 10^{-14}$ $\lambda = 5.0 \times 10^{-13}$ $\lambda = 1.0 \times 10^{-11}$

Fig. 5. Resulting images with (top) and without (bottom) adaptive regularization based on pixel-wise depth reliabilities

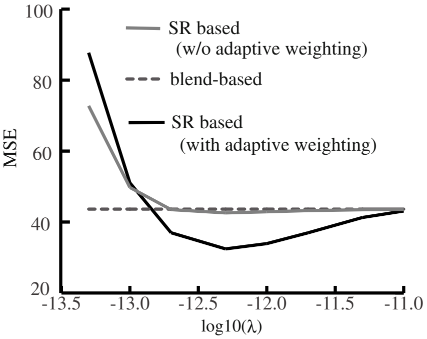


Fig. 6. Regularization factor vs. quality

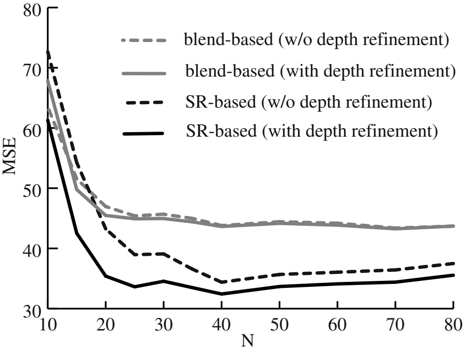


Fig. 7. Number of depths vs. quality

adaptive regularization we cannot obtain good results with any value of λ. The same results are shown quantitatively in Fig. 6. The horizontal axis denotes the value of λ in log scale, and the vertical axis is the mean squared error (MSE) against the ground truth image. The dashed line represents the quality of blend-based synthesis. The SR-based synthesis successfully improved the quality (reduced the MSE) if and only if the adaptive regularization was enabled.

Finally, we evaluated the performance change with regard to the number of candidate depths (N in Eq. (4)) and depth refinements (Eqs. (8) and (9)). The graph in Fig. 7 shows the relation between the number of candidate depths and the resulting image quality in MSE. As an overall trend, the quality improved as the number of depths increased, but using more than 40 depths had no benefit in our environment. As clearly seen in the graph, SR-based synthesis performed better than blend-based synthesis with a sufficient number of depths. Moreover, depth refinement was effective for improving the quality, especially when it was combined with SR-based synthesis.

6 Conclusion

We proposed a method for free-viewpoint image synthesis with resolution improvement. The main features of our method are its view-dependent approach focused on a given target viewpoint, fast and accurate semi-global depth estimation, and super-resolution-based synthesis with depth-reliability-based regularization. Experimental results validated the effectiveness of our method. Future work will be focused on its real-time implementation. Our current implementation with unoptimized MATLAB codes performs at an unsatisfactory speed. We plan to transplant it to C++ and CUDA codes to improve the processing rate.

Acknowledgments. This research is supported by the Strategic Information and Communication R&D Promotion Programs (SCOPE) of the Ministry of Internal Affairs and Communications, Japan.

A Appendix

A.1 Derivation of Mapping Function

Here we show how to derive the point correspondence between two cameras α and β with a known depth z. Let $P_{(\alpha)}$ be the 3×4 projection matrix of the camera α. An object point $\boldsymbol{X}$ is projected onto an image point $\boldsymbol{u}_\alpha$ as

$$\boldsymbol{p}_{(\alpha)} = P_{(\alpha)}\boldsymbol{X}, \text{ where } \boldsymbol{p}_{(\alpha)} = (u_\alpha, v_\alpha, 1)^t, \boldsymbol{X} = (X, Y, Z, 1)^t. \tag{17}$$

A plane located at $Z = z$ can be written as

$$[0, 0, 1, -z] \cdot \boldsymbol{X} = 0. \tag{18}$$

By combining Eqs. (17) and (18), we obtain

$$\left(\begin{array}{c} \boldsymbol{p}_{(\alpha)} \\ \hline 0 \end{array}\right) = \hat{P}_{(\alpha)}\boldsymbol{X}, \text{ where } \hat{P}_{(\alpha)} = \left(\begin{array}{c} P_{(\alpha)} \\ \hline 0\ 0\ 1\ -z \end{array}\right). \tag{19}$$

Similarly, we can also derive $\hat{P}_{(\beta)}$ for the camera β. By using them, we obtain the point correspondence between the two cameras as

$$\left(\begin{array}{c} \boldsymbol{p}_{(\beta)} \\ \hline 0 \end{array}\right) = \hat{P}_{(\beta)}\hat{P}_{(\alpha)}^{-1} \left(\begin{array}{c} \boldsymbol{p}_{(\alpha)} \\ \hline 0 \end{array}\right). \tag{20}$$

This is equivalent to the mapping function $P_{\alpha\to\beta}(\boldsymbol{u}_{\boldsymbol{\alpha}}, z)$ in Eq. (6).

A.2 Derivation of Depth Refinement Procedure

Assume a parabolic function, $y = ax^2 + bx + c$, to locally approximate the energy function $S(\boldsymbol{p}, n)$ around $n = D(\boldsymbol{p})$. We substitute three points, $(x, y) = (D(\boldsymbol{p}),\ S_{\min}(\boldsymbol{p}))$, $(D(\boldsymbol{p}) - 1,\ S_{\min}^{pre}(\boldsymbol{p}))$, and $(D(\boldsymbol{p}) + 1,\ S_{\min}^{next}(\boldsymbol{p}))$, to obtain the coefficients a, b, and c. This function clearly takes the minimum $c - b^2/4a$ at $x = -b/2a$, which are equivalent to Eqs. (8) and (9), respectively.

A.3 Derivation of warping function

A pseudo-code of the function $f_{t_\uparrow \to m}$ in Eq. (12) is given as follows.

```
00: function $I'_{(m)} = f_{t_\uparrow \to m}(I_{(t)}^{SR},\ \hat{D}_\uparrow)$
01:
02: for each $m$
03:     $D_{(m)} = \text{depth\_warping}(\hat{D}_\uparrow,\ t_\uparrow \to m)$
04: end
05:
06: $I'_{(m)}(\boldsymbol{p}) = 0$ for all $\boldsymbol{p} \in I'_{(m)}$
07: for each $\boldsymbol{p} \in I_t^{SR}$
08:     $\boldsymbol{p}_{(m)} = P_{t_\uparrow \to m}(\boldsymbol{p}, z_{\hat{D}_\uparrow(\boldsymbol{p})})$
09:     get integer pixel positions $\boldsymbol{p}_{(m),i}$ $(i = 1, 2, \ldots)$ around $\boldsymbol{p}_{(m)}$
10:     for each $\boldsymbol{p}_{(m),i}$
11:         if $||D_{(m)}(\boldsymbol{p}_{(m),i}) - \hat{D}_\uparrow(\boldsymbol{p})| \leq 1$
12:             get $r_i$ based on $|\boldsymbol{p}_{(m)} - \boldsymbol{p}_{(m),i}|$ and PSF
13:             $I'_{(m)}(\boldsymbol{p}_{(m),i}) = I'_{(m)}(\boldsymbol{p}_{(m),i}) + r_i I_{(t)}^{SR}(\boldsymbol{p})$
14:         end
15:     end
16: end
```

In lines 02–04, depth maps viewed from input viewpoints $D_{(m)}$ are obtained by warping $\hat{D}_\uparrow$ to the input viewpoints, whose details are given later. In line 06, all pixels of $I'_{(m)}$ are initialized with zero. In line 08, a pixel on $I_{(t)}^{SR}$, $\boldsymbol{p}$, is warped onto the m-th input camera, resulting in $\boldsymbol{p}_{(m)}$. Since $\boldsymbol{p}_{(m)}$ is not an integer pixel position in general, neighboring integer pixels $\boldsymbol{p}_{(m),i}$ are selected. After the occlusion test in line 11, we determine the contribution weight r_i in line 12. This weight is calculated from the distance between $\boldsymbol{p}_{(m)}$ and $\boldsymbol{p}_{(m),i}$, and the shape of the point spreading function (PSF). We use a box-shaped PSF that is equal to the pixel in size. In line 13, $I_{(t)}^{SR}(\boldsymbol{p})$ is weighted by r_i and added to $I'_{(m)}(\boldsymbol{p}_{(m),i})$. These procedures are iterated for every pixel $\boldsymbol{p} \in I_t^{SR}$.

The function depth_warping() is given as follows. In line 02, each pixel is initialized with 0, which corresponds to the infinite distance. In line 04, each pixel on $\hat{D}_\uparrow$ is warped to the m-th input viewpoint. Depth values of $D_{(m)}$ are updated with the occlusion test as shown by lines 05–07.

```
function $D_{(m)} = \text{depth\_warping}(\hat{D}_{\uparrow},\ t_{\uparrow} \rightarrow m)$

$D_{(m)}(\boldsymbol{p}) = 0$ for all $\boldsymbol{p} \in D_{(m)}$
for each $\boldsymbol{p} \in I_t^{SR}$
    $\boldsymbol{p}_{(m)} = \text{round}(P_{t_{\uparrow} \rightarrow m}(\boldsymbol{p},\ z_{\hat{D}_{\uparrow}(\boldsymbol{p})}))$
    if $D_{(m)}(\boldsymbol{p}_{(m)}) \leq \hat{D}_{\uparrow}(\boldsymbol{p})$
        $D_{(m)}(\boldsymbol{p}_{(m)}) = \hat{D}_{\uparrow}(\boldsymbol{p})$
    end
end
```

References

1. Kubota, A., et al.: Multiview Imaging and 3DTV. IEEE Signal Processing Magazine 24(6), 10–111 (2007)
2. Taguchi, Y., Koike, T., Takahashi, K., Naemura, T.: TransCAIP: A Live 3D TV System Using a Camera Array and an Integral Photography Display with Interactive Control of Viewing Parameters. IEEE Trans. Visualization and Computer Graphics 15(5), 841–852 (2009)
3. Park, S.-C., et al.: Super-resolution Image Reconstruction: A Technical Overview. IEEE Signal Processing Magazine 20(3), 21–36 (2003)
4. Matusik, W., Buehler, C., Raskar, R., Gortler, S.-J., McMillan, L.: Image-Based Visual Hulls. In: Proc. ACM SIGGRAPH, pp. 369–374 (2000)
5. Yang, R., Welch, G., Bishop, G.: Real-Time Consensus-Based Scene Reconstruction Using Commodity Graphics Hardware. In: Proceedings of Pacific Graphics, pp. 225–235 (2002)
6. Hirschmueller, H.: Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In: IEEE CVPR, pp. 807–814 (2005)
7. Hirschmuller, H.: Stereo Processing by Semiglobal Matching and Mutual Information. In: IEEE TPAMI, vol. 30(2), pp. 328–341 (2008)
8. Tung, T., Nobuhara, S., Matsuyama, T.: Simultaneous Super-Resolution and 3D Video Using Graph-Cuts. In: IEEE CVPR, pp. 1–8 (2008)
9. Goldluecke, B., Cremers, D.: Superresolution Texture Maps for Multiview Reconstruction. In: IEEE ICCV, pp. 1677–1684 (2009)
10. Mudenagudi, U., Gupta, A., Goel, L., Kushal, A., Kalra, P., Banerjee, S.: Super Resolution of Images of 3D Scenecs. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 85–95. Springer, Heidelberg (2007)
11. Takahashi, K., Ishii, M., Naemura, T.: Super-Resolution Plane Sweeping for Free-Viewpoint Image Synthesis. In: IEEE ICIP, pp. 2013–2016 (2011)
12. http://vision.middlebury.edu/stereo/

Heat Kernel Smoothing via Laplace-Beltrami Eigenfunctions and Its Application to Subcortical Structure Modeling

Seung-Goo Kim[1], Moo K. Chung[1,2,3,⋆] Seongho Seo[1],
Stacey M. Schaefer[3], Carien M. van Reekum[5], and Richard J. Davidson[3,4]

[1] Department of Brain and Cognitive Sciences,
Seoul National University, Korea
[2] Department of Biostatistics and Medical Informatics
[3] Waisman Laboratory for Brain Imaging and Behavior
[4] Department of Psychology and Psychiatry,
University of Wisconsin, Madison, WI, USA
[5] Centre for Integrative Neuroscience and Neurodynamics, School of Psychology
and Clinical Language Sciences, University of Reading, UK

Abstract. We present a new subcortical structure shape modeling framework using heat kernel smoothing constructed with the Laplace-Beltrami eigenfunctions. The cotan discretization is used to numerically obtain the eigenfunctions of the Laplace-Beltrami operator along the surface of subcortical structures of the brain. The eigenfunctions are then used to construct the heat kernel and used in smoothing out measurements noise along the surface. The proposed framework is applied in investigating the influence of age (38-79 years) and gender on amygdala and hippocampus shape. We detected a significant age effect on hippocampus in accordance with the previous studies. In addition, we also detected a significant gender effect on amygdala. Since we did not find any such differences in the traditional volumetric methods, our results demonstrate the benefit of the current framework over traditional volumetric methods.

1 Introduction

The amygdala and hippocampus are primary subcortical structures involved in emotion and memory [1,2]. Age and gender could be major factors that affect the functions and structures of these regions, as implied by postmortem studies [3]. Although the atrophy of brain tissues associated with the increase of age is reported in several hundreds subjects [4,5], the findings on the atrophy of amygdalar and hippocampal structures are somewhat inconsistent. The volume reduction of amygdala and hippocampus due to aging has been found in some studies [6,7,8], while other studies did not find such association [4,5,9,10]. For

[⋆] Corresponding author: Moo K. Chung, Waisman Center #281, 1500 Highland Ave. Madison, WI. 53705, mkchung@wisc.edu, http://www.stat.wisc.edu/~mchung/

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 36–47, 2011.

the effect of gender, one study reported significant differences in amygdala and hippocampus volume between the groups [11] whereas others failed to reproduce these [12]. The inconsistency between these results may have been due to the different image processing and analysis pipelines used in these studies.

In these volumetric studies, the total volume of the amygdala or hippocampus was typically estimated by tracing the region of interest (ROI) manually and counting the number of voxels within the ROI. The limitation of this ROI-based volumetry is that it cannot determine if the volume difference is diffuse over the whole ROI or localized within specific regions of the ROI [13]. Our proposed deformation-based morphometry (DBM) framework can localize the volume difference up to the mesh resolution at each surface mesh vertex.

Using the 3D deformation field derived from spatial normalization of MRI, we can model how the surfaces of subcortical structures are different from each other at the vertex level. Since the deformation field is noisy, it is necessary to smooth out the field along the surface to increase the signal-to-noise ratio (SNR). Further, smoothing is desirable in satisfying the assumptions of the random field theory (RFT), which is used in correcting for multiple comparisons [14,15]. For RFT to work, the Gaussianness and smoothness of data are needed [14,16]. As the amount of smoothing increases, Gaussianness and smoothness of data increases. With these motivations, we present a new framework of smoothing scalar and vector measurements using *heat kernel smoothing*, which is equivalent to performing isotropic diffusion but without discretizing the diffusion equation. The proposed framework is then used in examining the effect of age and gender on amygdala and hippocampus, contrasting the traditional volumetric analysis.

2 Method

We analyze the shape of subcortical structures as follows: (1) obtain a population mean volume by averaging the spatially normalized binary masks, and extract a template surface from the averaged binary volume (section 2.1), (2) interpolate the 3D displacement vector field onto the vertices of the surface meshes (section 2.1), (3) perform heat kernel smoothing on the displacement length along the template surface to reduce noise, and on the surface coordinates to smooth out the surface itself for better visualization (section 2.2 and 2.3), (4) apply a general linear model testing the effect of age and gender. The detailed description of each step is given in section 2 except the statistical inference which is given in section 3.

2.1 Images and Preprocessing

We have high resolution T1-weighted inverse recovery fast gradient echo anatomical 3D images, collected in 124 contiguous 1.2-mm axial slices (TE=1.8 ms; TR=8.9 ms; flip angle = 10°; FOV = 240 mm; 256 × 256 data acquisition matrix) of 69 middle age and elderly adults ranging between 38 to 79 years

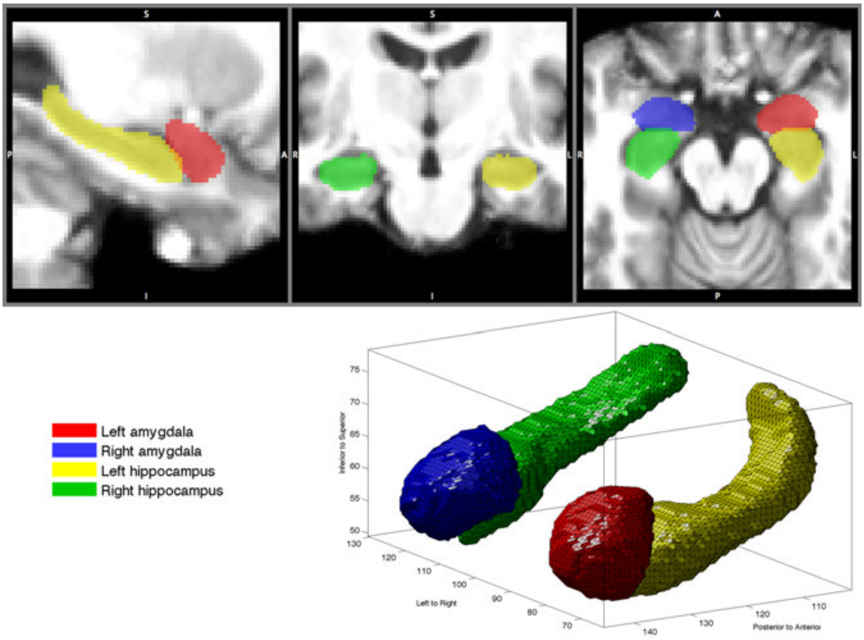


Fig. 1. Subcortical masks superimposed on MRI (top) and the corresponding isosurfaces of the masks (bottom)

(mean age = 58.04 ± 11.34 years). The data were originally collected for a national study for the health and well-being in the aged population, called MIDUS (Midlife in US; `http://midus.wisc.edu/`).

There are 23 males and 46 females. The amygdalae and hippocampi were manually segmented by a trained individual rater. Brain tissues in the MRI scans were first segmented using Brain Extraction Tool (BET) [17]. Then we performed a nonlinear image registration using the diffeomorphic shape and intensity averaging technique with cross-correlation as similarity metric through Advanced Normalization Tools (ANTS) [18]. A study-specific unbiased template was constructed from a random subsample of 10 subjects. Using the deformation field of warping the individual brain to the template, we deformed the amygdala and hippocampus binary masks to the template space. The normalized masks were then averaged to produce the subcortical masks. The isosurfaces of the subcortical masks are extracted using the marching cube algorithm [19]. The subcortical masks and the corresponding surfaces are shown in Fig. 1.

Using ANTS, we have the deformation vector field of warping an individual brain to the template. The vector field is defined on voxels. On the other hand, the vertices of the subcortical surface meshes are located within voxels. So we simply assigned the vector field onto the mesh vertices by linear interpolation (Fig. 2). The length of the displacement vector at each vertex is computed and used as a feature to measure the local shape variation.

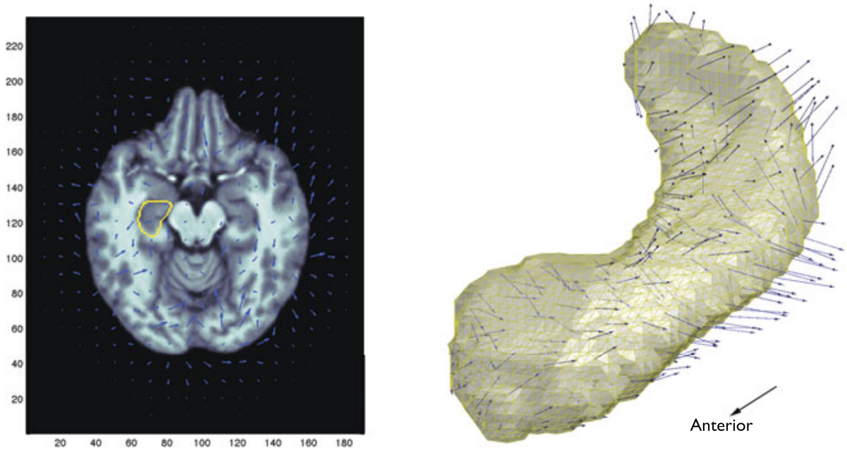


Fig. 2. Displacement vector field (blue arrows) of a subject on an axial slice of the template brain (left). Yellow contour in the left panel is the boundary of the left hippocampus in the template. The vector field has been interpolated on the left hippocampus surface (right).

2.2 Heat Kernel Smoothing

Since the displacement length on the template surface is noisy, it is necessary to smooth out the measurements to increase the signal-to-noise ratio (SNR) and to improve the smoothness and Gaussianness of data for RFT-based statistiscal inference [20]. We propose a new diffusion smoothing framework that uses the Laplace-Beltrami eigenfunctions.

Diffusion equations have been widely used in image processing as a form of noise reduction starting with Perona and Malik in 1990 [21]. Although numerous techniques have been developed for surface fairing and mesh regularization [20,22,23,24,25,26] based on heat diffusion. Most diffusion smoothing approaches mainly use finite element or finite difference schemes which is known to suffer numerical instability if the forward Euler scheme is used.

In this paper, we propose a new smoothing framework that constructs the heat kernel analytically using the eigenfunctions of the Laplace-Beltrami operator. Although solving the eigenfunctions of the Laplace-Beltrami operator requires the finite element method, the proposed method is analytic in a sense that heat kernel smoothing is formulated as a series expansion explicitly. We are not claiming our framework to be analytic which is theoretically impossible when dealing with real data. The proposed method represents isotropic heat diffusion analytically as a series expansion so it avoids the numerical instability associated with solving the diffusion equations numerically [20,22,27]. Our framework is an improvement over previous approaches in the sense that it bypasses the various numerical problems that are associated with previous approaches including numerical instability, slow convergence, and accumulated linearization error.

Consider a real-valued functional measurement $Y(p)$ defined on a manifold $\mathcal{M} \subset \mathbb{R}^3$. We assume the following additive model:

$$Y(p) = \theta(p) + \epsilon(p), \tag{1}$$

where $\theta(p)$ is the unknown mean signal to be estimated and $\epsilon(p)$ is a zero-mean Gaussian random field. We may assume $Y \in L^2(\mathcal{M})$, the space of square integrable functions on $\mathcal{M}$ with the inner product

$$\langle f, g \rangle = \int_{\mathcal{M}} f(p)g(p)d\mu(p),$$

where μ is the Lebesgue measure such that $\mu(\mathcal{M})$ is the volume of $\mathcal{M}$. Solving

$$\Delta\psi_j = \lambda_j \psi_j, \tag{2}$$

for the Laplace-Beltrami operator Δ on $\mathcal{M}$, we find the eigenvalues λ_j and eigenfunctions ψ_j. The eigenfunctions ψ_j form an orthonormal basis in $L^2(\mathcal{M})$ [28]. We may order eigenvalues as $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \cdots$ and corresponding eigenfunctions as $\psi_0, \psi_1, \psi_2, \cdots$.

heat kernel $K_\sigma(p, q)$ is then analytically given as

$$K_\sigma(p, q) = \sum_{j=0}^{\infty} e^{-\lambda\sigma} \psi_j(p)\psi_j(q), \tag{3}$$

where σ is the bandwidth of the kernel [29]. *Heat kernel smoothing* of Y is given analytically defined as

$$K_\sigma * Y(p) = \sum_{j=0}^{\infty} e^{-\lambda\sigma} \beta_j \psi_j(p), \tag{4}$$

where $\beta_j = \langle Y, \psi_j \rangle$ are Fourier coefficients. The heat kernel smoothing (4) is taken as an estimate for the unknown signal θ. Since the expansion (4) is a unique solution to isotropic heat diffusion, we can avoid the need to solve the diffusion using less stable numerical schemes such as the finite difference method [29,30].

2.3 Numerical Implementation

As the closed form expression for the eigenfunctions of the Laplace-Beltrami operator on an arbitrary curved surface is unknown, the eigenfunctions are numerically calculated by discretizing the Laplace-Beltrami operator. To solve the eigensystem (2), we need to discretize it on a triangular mesh using the Cotan discretization [31,32]. Using the Cotan discretization, (2) is linearized as the generalized eigenvalue problem:

$$\mathbf{C}\psi = \lambda \mathbf{A} \boldsymbol{\psi} \tag{5}$$

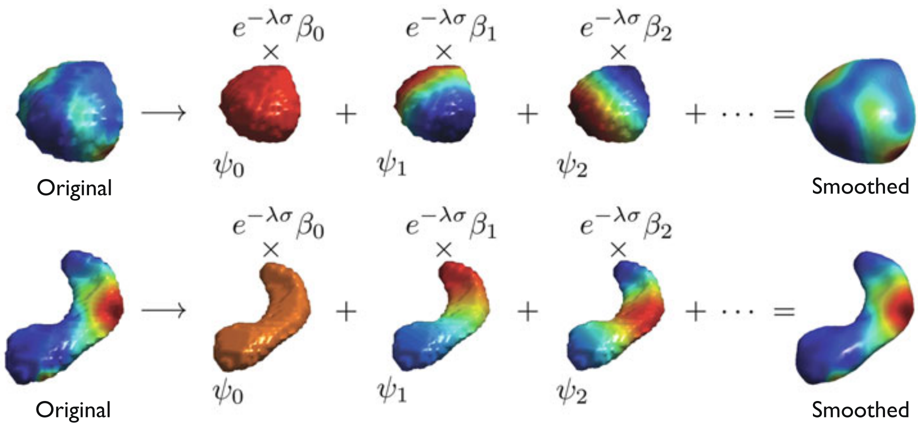

$e^{-\lambda\sigma}\beta_0$
×
$e^{-\lambda\sigma}\beta_1$
×
$e^{-\lambda\sigma}\beta_2$
×
Original
ψ_0
ψ_1
ψ_2
Smoothed
$e^{-\lambda\sigma}\beta_0$
×
$e^{-\lambda\sigma}\beta_1$
×
$e^{-\lambda\sigma}\beta_2$
×
Original
ψ_0
ψ_1
ψ_2
Smoothed

Fig. 3. Illustration of heat kernel smoothing. By summing the Laplace-Beltrami eigenfunctions, we smooth out functional measurements on surfaces. The left most surfaces are the noisy original surfaces with the displacement length. First three eigenfunctions ψ_0, ψ_1, ψ_2 are shown in the middle. The right most surfaces are the results of summation with $\sigma = 0.5$.

where $\mathbf{C}$ is the stiffness matrix, $\mathbf{A}$ is the mass matrix and $\boldsymbol{\psi} = (\psi(p_1), \cdots, \psi(p_n))'$ is the unknown eigenfunction evaluated at n mesh vertices. A first few eigenfunctions for the subcortical surfaces are shown in Fig. 3.

In this study, we have chosen the bandwidth σ=0.5 and used the finite eigenfunction expansion using up to 1,000 basis (Fig. 3). We smoothed the length of displacement vector field and the coordinates of template surfaces as well.

Once we obtained the basis functions ψ_j numerically, we need to estimate the Fourier coefficients β_j. It can be shown that the Fourier coefficients can be estimated as

$$\beta_j = \mathbf{Y}'\mathbf{A}\boldsymbol{\psi}_j, \tag{6}$$

where $\mathbf{Y} = (Y(p_1), \cdots, Y(p_n))'$ and $\boldsymbol{\psi}_j = (\psi_j(p_1), \cdots, \psi_j(p_n))'$ [33].

The MATLAB code for computing the eigenfunctions and performing heat kernel smoothing is available at `http://brainimaging.waisman.wisc.edu/~chung/lb/`.

2.4 Validation

The heat kernel smoothing framework is validated on a unit sphere where the Laplace-Beltrami eigenfunctions are exactly given as spherical harmonics. We used a spherical mesh with 40,962 uniformly sampled mesh vertices. Let Y_{lm} be the spherical harmonic of degree l and order m [34]. Due to the multiplicity, there are $2l$+1 eigenfunctions $Y_{l,-l}, \cdots, Y_{l,l}$ corresponding to the same eigenvalue $l(l+1)$. Further, any linear combination $\sum_{m=-l}^{l} \beta_{lm} Y_{lm}$ is an eigenfunction as well. So it is not possible to validate the accuracy of the obtained eigenfunctions. Therefore, we only checked if solving (5) produces the expected eigenvalues.

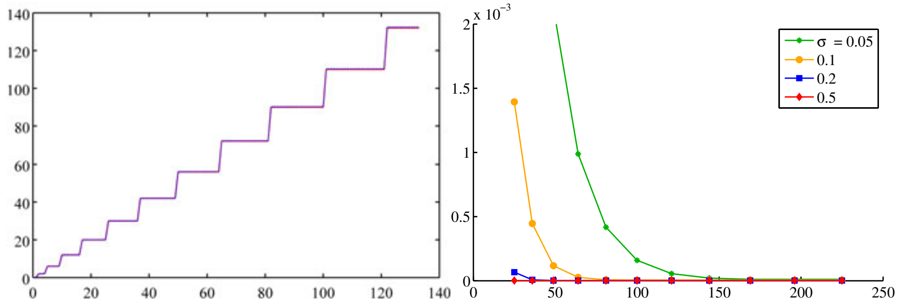


Fig. 4. Left: 133 eigenvalues are numerically computed (blue dotted) and compared against the ground truth (red solid) $\lambda_l = l(l+1)$ for up to degree $l = 11$. Right: The plot of the root mean squared errors (RMSE) of computed heat kernel over the number of eigenfunctions used (horizontal) for bandwidths 0.05, 0.1, 0.2 and 0.5. As the number of eigenfunctions increases, our implementation converge to the ground truth.

Fig. 4 shows the 133 computed eigenvalues compared against the ground truth. The maximum possible relative error is 0.0032 (0.32%).

We also checked the accuracy of the constructed heat kernel. On a unit sphere, the heat kernel is given by

$$K_\sigma(p,q) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} e^{-l(l+1)\sigma} Y_{lm}(p) Y_{lm}(q). \tag{7}$$

We have taken the degree $l = 85$ expansion as the ground truth and compared it to the numerically constructed heat kernel. The RMSE of heat kernel against the ground truth was computed for various bandwidth between 0.05 and 0.5 (Fig. 4). The rate of convergence depends on the bandwidth. As the number of eigenfunctions increases, the constructed heat kernel converge to the ground truth quickly. Beyond 150 eigenfunctions, the reconstruction error is negligible.

3 Results: General Linear Models on Surface Shapes

3.1 Traditional Volumetric Analysis

In the traditional volumetric approach, the volumes of amygdala and hippocampus binary mask were simply computed by counting the number of voxels within the mask. In order to account for the effect of intersubject variability in brain size, the brain volume excluding cerebellum was computed and covariated in general linear models.

The brain volume is significantly correlated with the amygdala (left: r= 0.55, $p < 10^{-5}$; right: r=0.49, $p < 10^{-4}$) and the hippocampus volumes (left: r= 0.59, $p < 10^{-7}$; right: r=0.63, $p < 10^{-8}$). Since amygdala and hippocampus volumes are dependent on the whole brain volume, we really need to factor out the brain volume in the general liner models.

We model the $Volume$ of amygdala and hippocampus as

$$Volume = \beta_1 + \beta_2 \cdot Brain + \beta_3 \cdot Age + \beta_4 \cdot Gender + \epsilon \tag{8}$$

where ϵ is zero mean Gaussian noise and $Brain$ is the total brain volume. The age and gender effects are determined by testing the significance of parameters β_3 and β_4 at $\alpha = 0.01$. The results are displayed in Figure 5.

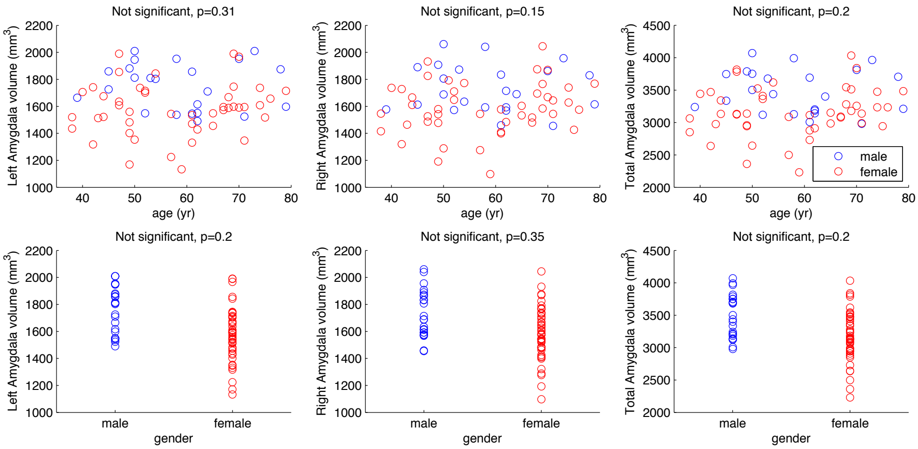


Fig. 5. Scatterplots of left, right and combined amygdala volumes over age (top) and gender (bottom)

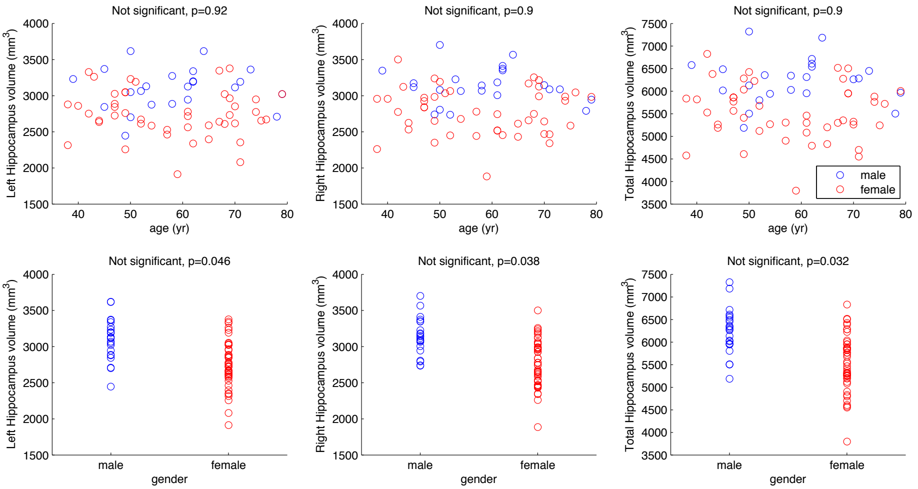


Fig. 6. Scatterplots of the volume of left, right and total hippocampus over age (top) and gender (bottom)

For the amygdala volume, we did not find a significant effect of age (left p= 0.31; right p= 0.15; combined p= 0.20) nor gender (left p= 0.20; right p= 0.35; combined p= 0.25) For the hippocampus volume, we did not find a significant effect of age (left p= 0.92; right p= 0.90; total p= 0.90) nor gender (left p= 0.05; right p= 0.04; total p= 0.03).

Since our results are based on the volume of the whole amygdala and hippocampus, it is still unclear if there are any localized shape differences within the parts of amygdala and hippocampus.

3.2 Localized Subcortical Shape Analysis

The length of displacement vector fields along the template surfaces were computed and smoothed as described in section 2. Then *Length* is regressed over the total brain volume and other variables:

$$Length = \beta_1 + \beta_2 \cdot Brain + \beta_3 \cdot Age + \beta_4 \cdot Gender + \epsilon \qquad (9)$$

where ϵ is zero mean Gaussian noise. The age and gender effects are determined by testing the significance of parameters β_3 and β_4 at $\alpha = 0.01$. We used SurfStat MATLAB toolbox (`http://galton.uchicago.edu/faculty/InMemoriam/worsley/research/surfstat/`), for the statistical analysis and multiple comparison correction. The details on the SurfStat package is given in [34]. The results are displayed in Figure 7.

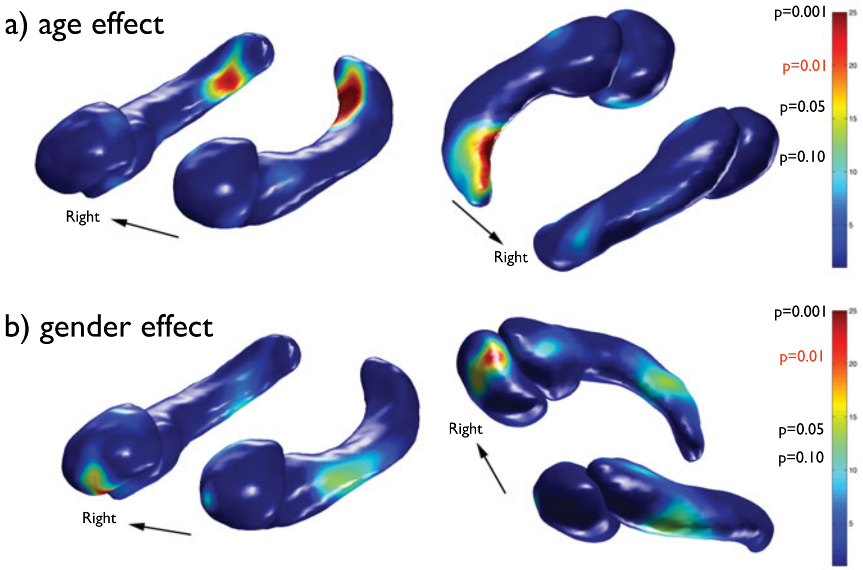


Fig. 7. F-statistic maps on the amygdala and hippocampus surfaces showing the age (a) and gender (b) effects with corresponding p-values indicated. The posterior regions of the both left and right hippocampi show a significant age effect. The ventral region of the right amygdala shows a significant gender effect.

Age effect. We found the region of significant effect of age on the posterior part of hippocampi (left: max F = 39.43, $p < 10^{-5}$; right: max F = 23.11, $p = 0.002$) Particularly, on the caudal regions of the left and right hippocampi, we found highly localized signals. It is consistent with other shape modeling studies on hippocampus [35,36]. We did not find any age effects on the amygdala surface at $\alpha = 0.01$.

Gender effect. We found a highly focalized region of gender effect on the inferior part of the right amygdala (max F = 24.66, $p < 0.001$). In particular, the gender effect is focused around the ventral part of laterobasal group [37].

We did not find any significant gender effects on the left amygdala and hippocampi.

4 Conclusion

We have presented a new subcortical structure shape modeling framework using heat kernel smoothing constructed with the Laplace-Beltrami eigenfunctions. The proposed framework demonstrated higher sensitivity in modeling shape variations compared to the traditional volumetric analysis. The ability to localize subtle morphological difference may provide an anatomical evidence for the functional organization within human subcortical structures.

References

1. LeDoux, J.: The amygdala. Current Biology 17(20), R868–R874 (2007)
2. Alvarez, P., Squire, L.R.: Memory consolidation and the medial temporal lobe: a simple network model. Proceedings of the National Academy of Sciences 91(15), 7041–7045 (1994)
3. Miller, A., Alston, R., Corsellis, J.: Variation with age in the volumes of grey and white matter in the cerebral hemispheres of man: measurements with an image analyser. Neuropathology and Applied Neurobiology 6(2), 119–132 (1980)
4. Good, C., Johnsrude, I., Ashburner, J., Henson, R., Friston, K., Frackowiak, R.: A voxel-based morphometric study of ageing in 465 normal adult human brains. NeuroImage 14(1), 21–36 (2001)
5. Walhovd, K., Westlye, L., Amlien, I., Espeseth, T., Reinvang, I., Raz, N., Agartz, I., Salat, D., Greve, D., Fischl, B., et al.: Consistent neuroanatomical age-related volume differences across multiple samples. Neurobiology of Aging (2009)
6. Bigler, E., Blatter, D., Anderson, C., Johnson, S., Gale, S., Hopkins, R., Burnett, B.: Hippocampal volume in normal aging and traumatic brain injury. American Journal of Neuroradiology 18(1), 11 (1997)
7. Du, A., Schuff, N., Chao, L., Kornak, J., Jagust, W., Kramer, J., Reed, B., Miller, B., Norman, D., Chui, H., et al.: Age effects on atrophy rates of entorhinal cortex and hippocampus. Neurobiology of Aging 27(5), 733–740 (2006)
8. Walhovd, K., Fjell, A., Reinvang, I., Lundervold, A., Dale, A., Eilertsen, D., Quinn, B., Salat, D., Makris, N., Fischl, B.: Effects of age on volumes of cortex, white matter and subcortical structures. Neurobiology of Aging 26(9), 1261–1270 (2005)

9. Sullivan, E., Marsh, L., Mathalon, D., Lim, K., Pfefferbaum, A.: Age-related decline in mri volumes of temporal lobe gray matter but not hippocampus. Neurobiology of Aging 16(4), 591–606 (1995)
10. Sullivan, E., Marsh, L., Pfefferbaum, A.: Preservation of hippocampal volume throughout adulthood in healthy men and women. Neurobiology of Aging 26(7), 1093 (2005)
11. Good, C., Johnsrude, I., Ashburner, J., Henson, R., Friston, K., Frackowiak, R.: Cerebral asymmetry and the effects of sex and handedness on brain structure: a voxel-based morphometric analysis of 465 normal adult human brains. NeuroImage 14(3), 685–700 (2001)
12. Gur, R.C., Gunning-Dixon, F., Bilker, W.B., Gur, R.E.: Sex differences in temporo-limbic and frontal brain volumes of healthy adults. Cerebral Cortex 12(9), 998–1003 (2002)
13. Chung, M.K., Worsley, K.J., Paus, T., Cherif, D.L., Collins, C., Giedd, J., Rapoport, J.L., Evans, A.C.: A unified statistical approach to deformation-based morphometry. NeuroImage 14, 595–606 (2001)
14. Adler, R.: On excursion sets, tube formulas and maxima of random fields. Annals of Applied Probability, 1–74 (2000)
15. Worsley, K., Marrett, S., Neelin, P., Vandal, A., Friston, K., Evans, A., et al.: A unified statistical approach for determining significant signals in images of cerebral activation. Human Brain Mapping 4(1), 58–73 (1996)
16. Worsley, K., Evans, A., Marrett, S., Neelin, P.: A three-dimensional statistical analysis for CBF activation studies in human brain. Journal of Cerebral Blood Flow and Metabolism 12, 900 (1992)
17. Smith, S.: Fast robust automated brain extraction. Human Brain Mapping 17(3), 143–155 (2002)
18. Avants, B., Epstein, C., Grossman, M., Gee, J.: Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. Medical Image Analysis 12(1), 26–41 (2008)
19. Lorensen, W., Cline, H.: Marching cubes: A high resolution 3D surface construction algorithm. ACM Siggraph Computer Graphics 21(4), 163–169 (1987)
20. Chung, M.K., Worsley, K.J., Robbins, S., Paus, T., Taylor, J., Giedd, J.N., Rapoport, J.L., Evans, A.C.: Deformation-based surface morphometry applied to gray matter deformation. NeuroImage 18, 198–213 (2003)
21. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)
22. Andrade, A., Kherif, F., Mangin, J., Worsley, K., Paradis, A., Simon, O., Dehaene, S., Le Bihan, D., Poline, J.B.: Detection of fmri activation using cortical surface mapping. Human Brain Mapping 12, 79–93 (2001)
23. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. IEEE Transactions on Image Processing 7(3), 310–318 (1998)
24. Malladi, R., Ravve, I.: Fast Difference Schemes for Edge Enhancing Beltrami Flow. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 343–357. Springer, Heidelberg (2002)
25. Tang, B., Sapiro, G., Caselles, V.: Direction diffusion. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1245–1252 (1999)
26. Taubin, G.: Geometric signal processing on polygonal meshes. In: Eurographics 2000 – State of the Art Report (2000)

27. Joshi, A., Shattuck, D., Thompson, P., Leahy, R.: A parameterization-based numerical method for isotropic and anisotropic diffusion smoothing on non-flat surfaces. IEEE Transactions on Image Processing 18(6), 1358–1365 (2009)
28. Lévy, B.: Laplace-beltrami eigenfunctions towards an algorithm that understands geometry. In: IEEE International Conference on Shape Modeling and Applications, SMI 2006, p. 13. IEEE (2006)
29. Chung, M., Robbins, S., Dalton, K., Davidson, R., Alexander, A., Evans, A.: Cortical thickness analysis in autism with heat kernel smoothing. NeuroImage 25(4), 1256–1265 (2005)
30. Tasdizen, T., Whitaker, R., Burchard, P., Osher, S.: Geometric surface smoothing via anisotropic diffusion of normals. In: IEEE Visualization Conference (2002)
31. Chung, M., Taylor, J.: Diffusion smoothing on brain surface via finite element method. In: IEEE International Symposium on Biomedical Imaging: Nano to Macro, pp. 432–435. IEEE (2004)
32. Qiu, A., Bitouk, D., Miller, M.: Smooth functional and structural maps on the neocortex via orthonormal bases of the laplace-beltrami operator. IEEE Transactions on Medical Imaging 25(10), 1296–1306 (2006)
33. Zhang, H., van Kaick, O., Dyer, R.: Spectral methods for mesh processing and analysis. In: Eurographics 2007–State of the Art Reports, pp. 1–22 (2007)
34. Chung, M.K., Worsley, K.J., Nacewicz, B.M., Dalton, K.M., Davidson, R.J.: General multivariate linear modeling of surface shapes using surfstat. NeuroImage 53(2), 491–505 (2010)
35. Qiu, A., Miller, M.: Multi-structure network shape analysis via normal surface momentum maps. NeuroImage 42(4), 1430–1438 (2008)
36. Xu, Y., Valentino, D., Scher, A., Dinov, I., White, L., Thompson, P., Launer, L., Toga, A.: Age effects on hippocampal structural changes in old men: the haas. NeuroImage 40(3), 1003–1015 (2008)
37. Amunts, K., Kedo, O., Kindler, M., Pieperhoff, P., Mohlberg, H., Shah, N., Habel, U., Schneider, F., Zilles, K.: Cytoarchitectonic mapping of the human amygdala, hippocampal region and entorhinal cortex: intersubject variability and probability maps. Anatomy and Embryology 210(5), 343–352 (2005)

SLAM and Navigation in Indoor Environments

Shang-Yen Lin and Yung-Chang Chen

Department of Electrical Engineering, National Tsing Hua University,
30013 Hsinchu, Taiwan
ycchen@ee.nthu.edu.tw

Abstract. In this paper, we propose a system for wheeled robot SLAM and navigation in indoor environments. An omni-directional camera and a laser range finder are the sensors to extract the point features and the line features as the landmarks. In SLAM and self-localization while navigation, we use extended Kalman filter (EKF) to deal with the uncertainty of robot pose and landmark feature estimation. After the map is built, robot can navigate in the environment based on it. We apply two scale path-planning for navigation. The large-scale planning finds an appropriate path from starting point to destination. The local-scale path-planning fills up the drawbacks of the prior step, such as dealing with the static and dynamic obstacles and smoothing the path for easier robot following. Through the experiment results, we show that the proposed system can smoothly and correctly locate itself, build the environment map and navigate in indoor environments.

Keywords: SLAM, EKF, navigation, path-planning, obstacle avoidance, robot.

1 Introduction

In recent years, there is more and more attention on robotics research, especially the service robot industry all over the world. Many applications of the robotic technology have been developed, such as security robots, nursing robots, and so on. All of them need to navigate in an associated environment and locate themselves. There are four major problems needed to be overcome, which are self-localization, environment map establishment, path to destination detection and collision avoidance. However, these four problems are not independent, but mutually correlated. For example, once self-localization consists in error, it may cause the wrong map building. And the wrong map will cause self-localization in larger error. Furthermore, the wrong map or self-estimated location may induce path-planning failure.

Fortunately, a number of researches have worked on these challenges. Simultaneous localization and mapping (SLAM) method builds an environment map using only relative environment observation and using the map for robot localization at the same time. An unbiased map needs correct robot location and vice versa. Smith et al. [1] [2] used probabilistic model, a state vector describing robot location and landmark position with a covariance matrix describing their mutual uncertainty, and extended Kalman filter (EKF) to represent and estimate the spatial uncertainty. Durrant-Whyte et al. [3] [4] proposed a framework of SLAM. Doucet et al. [5] solved the SLAM problem by means of Rao-Blackwellized particle filter (RBPF).

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 48–60, 2011.

Montemerlo et al. proposed FastSLAM algorithm which tracks the landmarks by extended Kalman filter and estimate the robot's pose by Rao-Blackwellized particle filter. There are also many researches using different sensors for SLAM [6][7][8].

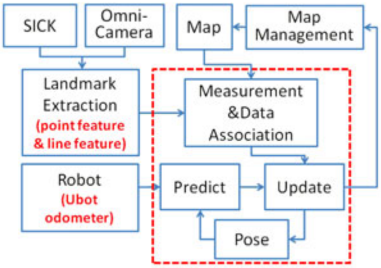
Also many researches discussed about path planning and obstacle avoidance. The path planning problems are typically approached using one of these two categories: search-based, sampling-based. The basic idea of search-based planning is using regular grid cells to represent the configuration space. The path planning problem is done by searching the grids and finding a point-to-point path from starting grid to the goal grid. Dijkstra [9] first proposed the Dijkstra's Algorithm, which solves the shortest path problem by breath-first search. A* algorithm [10] further uses an admissible heuristic to reduce the search region. The continuing improvements including D* algorithm [11] which makes re-plan more efficient, Anytime A* [12] which concerns the deliberation time and AD* [13] which combines D* and Anytime A*. The sampling based planning does not use the regular grid cells but samples the vertices in the configuration space with appropriate edge assignment between them, and finds path from the candidate vertices. The probabilistic roadmap (PRM) [14] generates vertex by random sampling. The rapidly-exploring random tree (RRT) [15] makes the sampling more efficient. There are also Anytime RRT [16] and Anytime Dynamic RRT [17] Algorithms proposed for improving the speed of planning and re-planning.

In this paper, we propose a system for wheeled robot navigation in indoor environments using only on-robot sensors. When the robot enters a new environment, we first build the environment map by SLAM (Simultaneous Localization and Mapping). In our approach, we choose the omni-directional camera and laser range finder as the on-robot sensors which have wide sensing field. The property of wide sensing field is very important for robot localization and obstacle capturing because of increasing the duration of landmarks and obstacles observation, and decreasing the effect of landmarks being covered by obstacles. After the environment map has been built, the robot can navigate by finding the appropriate path from the built map. We separate the robot navigation into two parts. The first part is the large-scale path-planning, which is similar as people select which path to go through. The other part is the local-scale path-planning for obstacle avoidance. This part rapidly generates a collision-free path and guarantees the robot real-time avoiding the obstacles when there are static or dynamic obstacles on the way to goal.

The remaining sections of the paper are organized as follows. Section 2 gives the overview of our system. Section 3 introduces feature extraction with omni-directional camera and laser range finder. The SLAM and localization method using the point and line features is presented in section 4. Section 5 presents the large scale and local scale path-planning for robot navigation. Experimental results are shown in Section 6. Finally, conclusion is presented in section 7.

2 System Overview

Figure 1 and Figure 2 show the flowcharts of our system. Figure 1 shows the SLAM flowchart, which is used to build the map when robot first enters a new environment. We utilize the omni-directional camera and SICK laser range finder to extract landmarks. The landmarks used in our system are point features and line features, which will be briefly described in next section. We use the extracted features and odometer data for SLAM. After data association, we revise error with extended Kalman filter.



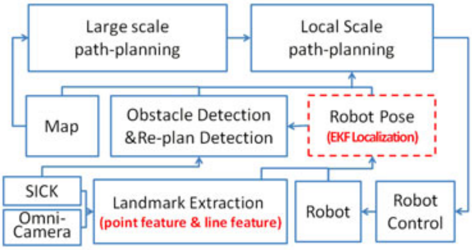

Large scale path-planning
Local Scale path-planning
Map
Obstacle Detection &Re-plan Detection
Robot Pose
(EKF Localization)
SICK
Omni-Camera
Landmark Extraction
(point feature & line feature)
Robot
Robot Control

Fig. 1. Flowchart of SLAM

Fig. 2. Flowchart of robot navigation

Figure 2 shows the flowchart of robot navigation. After the environment map is built, robot can navigate based on it. When robot navigates in the indoor environment, we still use the same method as Figure 1 for robot self-localization. The dotted block" Robot Pose" in Figure 2 is same as the dotted block in Figure 1. For navigation, we first apply a large-scale path-planning. In this step, the algorithm finds an appropriate path from the location of robot to the destination. After the path is generated, robot can go along it and move toward destination. While the robot is moving, there may be some obstacles which block the original path as detected by the laser range finder. The local-scale path-planning can quickly generate a new collision free path for avoiding robot collision. Finally, the robot can safely achieve to the destination.

3 Landmark Extraction

We use an omni-directional camera and a laser range finder as our sensor system to extract the landmarks in the environment. It is easy to combine data of these two sensors because they are center-aligned, facing the same direction, and both sensing data can be represented in polar coordinates. In landmark extraction, we extract two types of landmarks: the point landmarks and the line landmarks. The point landmarks are used for both x-y location and robot orientation estimation, similar to many point-feature-based SLAM works. But the estimation with only point landmarks may have larger error if there are very few number of point landmarks in the observation region. Therefore, we add the line landmarks in our system. The line landmarks can improve the accuracy of robot orientation estimation.

3.1 Point Landmarks

A point landmark is the 2-D position of a 3-D vertical line, which is the intersection of the 3-D line with x-y plane as shown in Figure 3(a). It is very efficient to extract the vertical lines from images captured by onmi-directional camera, because of the property that all of the 3-D vertical lines extending pass through the center. Figure

3(b) shows the omni-directional camera data input. Because of the ratio of the image could influence the bearing information of the extracted landmark, we first resize the image to equal ratio, as shown in Figure 3(c). To make our processing concentrate on the useful region, we use a mask as given in Figure 3(d). Only the data in the white region will be processed. Then we apply the Canny edge detector to find out the edges. After we got the edge points, we record how many edge points exist in each angle degree. Once the number of existing edge points in an angle degree exceeds a threshold, a vertical line is affirmed. Combining with the laser range finder data, as the yellow points shown in Figure 3(f), we can get the position of the point landmarks. Finally, to reduce the observation error, we ignore those landmarks which are too far away. And for those landmarks existing in continuous angle degree, we only use the two sides of them as our point landmarks.

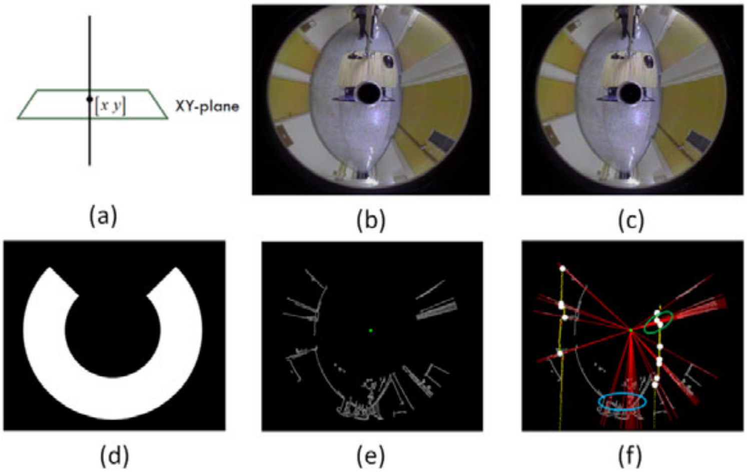


Fig. 3. Illustration of point landmarks extraction

3.2 Line Landmarks

The line landmarks are the horizontal straight lines in the environment, which are extracted from the laser range finder data. Figure 4(a) shows the laser range finder data input. For extracting the straight lines, we first find the break points for separating each straight line. For each range point s_i, if s_i satisfies one of the three conditions in (1), we consider s_i as a break point. These three conditions are considered for different situation. The first condition focuses on the distance between two continuous range points. If the distance is huge, as shown by red circles in Figure 4(b), two sides of s_i should not belong to same straight line. The second condition considers the angle between $\overrightarrow{s_{i-1}s_i}$ and $\overrightarrow{s_is_{i+1}}$. A large angle should not occur on a straight line. The third condition is dealing with the situation shown by the green circles in Figure 4(b). We can see s_i in two green circles in Figure 4(b) have similar information of distance and angle, but those on the left form a straight line while those on the right are break points. It is hard to use one threshold to distinguish them. Therefore, we use two smaller thresholds for double checking. After the break-point detection, those remaining range points could be seen as straight line points. For the sets consisting of consecutive points more than a threshold, apply the least square method and find the parameter of the line landmarks.

$$s_i \in \text{break point} \quad if \begin{cases} dis(s_{i-1},s_i) > \lambda_{d1} \;\|\; dis(s_i,s_{i+1}) > \lambda_{d1} \\ \left|dir(s_{i-1},s_i) - dir(s_i,s_{i+1})\right| > \lambda_{\theta 1} \\ \left(\begin{array}{c} (\lambda_{d1} > dis(s_{i-1},s_i) > \lambda_{d2} \;\|\; \lambda_{d1} > dis(s_i,s_{i+1}) > \lambda_{d2}) \\ \&\& \\ \left|dir(s_{i-1},s_i) - dir(s_i,s_{i+1})\right| > \lambda_{\theta 2} \end{array} \right) \end{cases}$$

$$where \quad \lambda_{d1} > \lambda_{d2} \;, \quad \lambda_{\theta 1} > \lambda_{\theta 2} \tag{1}$$

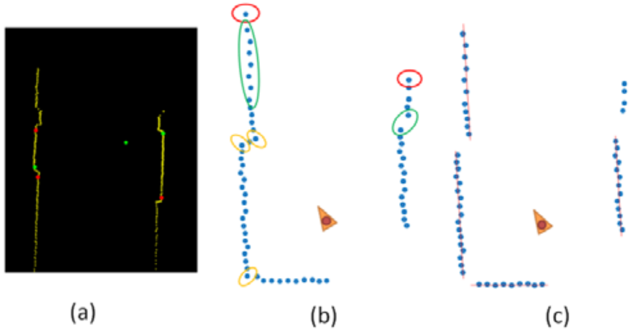

(a) (b) (c)

Fig. 4. Illustration of line landmarks extraction

4 Extended Kalman Filter SLAM

In SLAM, the uncertainty of the robot pose and landmark position is the dominant problem to be solved. In our system, we use EKF to handle uncertainties. The basic Kalman Filter is suited for a linear system. For a non-linear case we should linearize the original system appropriately. Generally, the EKF SLAM uses a state vector and a covariance matrix to describe status of the robot pose and landmark position. In our system the state vector is described as follows.

$$\mathbf{x}_k = [\mathbf{x}_r \quad m_1 \quad \ldots \quad m_n]^T \qquad \mathbf{x}_r = [x_r \quad y_r \quad \theta_r]^T \qquad \mathbf{m}_i = \begin{cases} [x_i \quad y_i]^T & \text{(point landmarks)} \\ [\theta_i] & \text{(line landmarks)} \end{cases} \tag{2}$$

$\mathbf{x}_k$ is the state vector at time k. $\mathbf{x}_r$ is the robot pose vector, where $[x_r \quad y_r]^T$ is the robot location in the world coordinate and θ_r is the orientation of the robot. $\mathbf{m}_i$ is the i^{th} landmarks as $[x_i \quad y_i]^T$ for point landmarks and $[\theta_i]$ for line landmarks.

- SLAM by EKF contains following main steps:

4.1 Prediction

We use the motion model and the previous state to predict the current state. The state prediction step is

$$\hat{\mathbf{x}}_{r\ k|k-1} = f(\hat{\mathbf{x}}_{r\ k-1|k-1}, \mathbf{u}_{k-1}) + q_k \tag{3}$$

$\mathbf{u}_{k-1}$ is the odometer data of motion, including velocity and angular velocity.

$$\mathbf{u}_k = \begin{bmatrix} V_k & \omega_k \end{bmatrix} \qquad \text{where } V = \frac{v_l + v_r}{2}, \omega = \frac{v_l - v_r}{l} \tag{4}$$

f is a non-linear function, describing how the robot moves in the world coordinate.

$$f(\hat{\mathbf{x}}_{r\ k-1|k-1}, \mathbf{u}_{k-1}) = \begin{bmatrix} x_{r\ k-1} + V_{k-1}\Delta T \cos(\theta_{r\ k-1}) \\ y_{r\ k-1} + V_{k-1}\Delta T \sin(\theta_{r\ k-1}) \\ \theta_{r\ k-1} + \Delta T\,\omega_{k-1} \end{bmatrix} \tag{5}$$

And the covariance matrix $P_{k-1|k-1}$ is propagated through the linearized state transition function f, yielding $P_{k|k-1}$ given by

$$P_{k|k-1} = J_f P_{k-1|k-1} J_f^T + Q_k \qquad \text{where } J_f \text{ is the Jacobian of } f \text{ at time } k \tag{6}$$

4.2 Observation

The observation equation for i^{th} landmark can be written as

$$z_i = h_i(x_{k-1} \mid x_{k-1}) + w_i \tag{7}$$

where w_i is the uncertainty of observation which is temporally uncorrelated and zero-mean random noise. For point landmarks, we use both range and bearing information. For line landmarks, we only use the bearing information. The following are their measurement functions.

- Point landmarks:

$$z_i = \begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2} \\ arcTan2(y_i - y_r, x_i - x_r) - \theta_r \end{bmatrix} + w_i \tag{8}$$

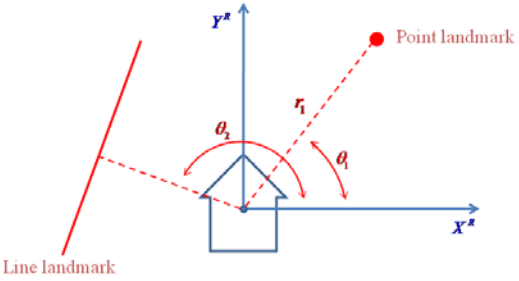


Fig. 5. Relation between landmarks and robot

- Line landmarks:

$$z_i = normal\_vector(line_i) + w_i \tag{9}$$

4.3 Data Association

For point landmarks, we use the Mahalanobis distance, which makes the distance error measurement take the correlation of the data set into account. If the Mahalanobis distance between the observed landmark z and the recorded landmark h_i is smaller than the threshold γ, we determine that z is associated with h_i.

$$v_i S_i^{-1} v_i^T < \gamma \qquad where \quad v_i = z - h_i \quad , \quad S_i = J_{h_i} P J_{h_i}^T + R \tag{10}$$

For line landmarks, we use the bearing information and the distance of the center of robot to feature line for association.

$$|\theta - \theta_i| < \gamma_\theta \qquad |d - d_i| < \gamma_d \tag{11}$$

4.4 Update

After landmark extraction and association, the measurement residual of associated landmarks can be used for EKF update. The Kalman gain K_k is computed as

$$\mathrm{K}_k = \mathrm{P}_{k|k-1} J_{hk}^T / \mathrm{S}_k \qquad where \quad J_{hk} = \frac{\partial h}{\partial \mathrm{x}_k}, \quad \mathrm{S}_k = J_{hk} \mathrm{P}_{k|k-1} J_{hk}^T + \mathrm{R}_{H_k} \tag{12}$$

The state vector and covariance are updated as

$$\begin{aligned} \mathrm{P}_{k|k} &= (\mathrm{I} - \mathrm{K}_k J_{hk}^T) \mathrm{P}_{k|k-1} \\ \hat{\mathrm{x}}_k &= \hat{\mathrm{x}}_{k|k-1} + \mathrm{K}_k v_k \qquad where \quad v_k = z_k - h_k(\hat{\mathrm{x}}_{k|k-1}) \end{aligned} \tag{13}$$

5 Path Planning for Robot Navigation

The robot navigation consists of two path planning parts. The first part is the large-scale path planning, which is similar to people choosing the appropriate path for walking along to the destination. In our system, we apply a search based method A* with big grid size. The big grid size planning has rough results but makes the plan accomplished rapidly. In the first step we do not really need a precise path because the next step deals with the obstacle avoidance. The second step, local-scale path planning, is composed of an orientation decision method and a RRT-based path planning. The path planning in this step can rapidly generate a substitute path for path smoothing or collision avoidance. The orientation decision guarantees a real-time command for reducing risks, even if the planning is not achieved in deliberation time.

5.1 Large-Scale Path Planning

In large-scale path planning, we use A* algorithm to find the path from the pre-built environment map, as shown in figure 7. A* is a best-first search using a heuristic cost function $f(x) = g(x) + h(x)$, where $g(x)$ is the cost from the starting node to the current node, and $h(x)$ is the heuristic estimation of the cost from the current node to the destination.

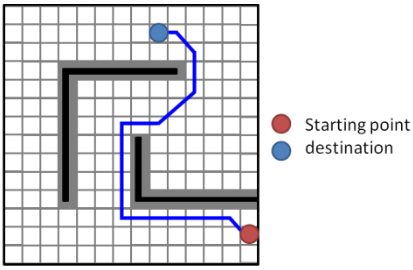


Fig. 6. Large-scale path planning by A*

5.2 Local-Scale Path Planning for Obstacle Avoidance

The local-scale path planning takes place in two situations. When the original planned path is blocked by any obstacles, re-planning applied for a collision-free path. When the original path is including extremely sharp turning angle, re-planning is applied for a smooth path.

- **RRT-based path planning.**

```
GrowTree(tree)
    while($x_{goal}$ != $x_{new}$)
      $x_{target}$ = GenerateTarget();
      $x_{nearest}$ = NearestNeighbor($x_{target}$, tree);
      $x_{new}$ = Extend($x_{nearest}$, $x_{target}$);
      if(CollisionCheck($x_{new}$));
        tree.add($x_{new}$);
GenerateTarget()
    $p$ = RandInt()%100;
    if($p$ < $\lambda_g$)
      return($x_{goal}$)
    else
      return RandomPoint();
Main()
    tree.init($x_{start}$);
    GrowTree(tree);
```

Fig. 7. The RRT Algorithm

The standard RRT algorithm is shown in Figure 7. To grow the tree, first x_{target} is randomly sampled from the configuration space by function **GenerateTarget**. To make the tree grow more efficiently and focused on x_{goal}, **GenerateTarget** returns x_{goal} with probability Λ_g. Then, the **NearestNeighbor** function finds $x_{nearest}$, which is the tree node closest to x_{target}. After that, a new node is generated by **Extend** function. If the new node is free from collision, add the new node in. Else, no extension applied. These steps are repeated until x_{goal} is reached.

In our system, we grow the RRT in three dimensional space, including the 2-D location x, y and the robot orientation θ. θ is used for smooth path generation. The **NearestNeighbor** function considers (14) as the distance between x_i and x_{target}.

$$\left|\overrightarrow{x_i\ x_{\text{target}}}\right| + \omega\cdot\left(dir(\overrightarrow{x_i\ x_{\text{target}}}) - orientation(x_i)\right) \tag{14}$$

,where ω is the weighting const.

The **Extend** function also needs to consider the orientation. For the smoothness of the generated path, x_{new} can only turn θ_{th} even if the orientation difference between x_i and $\overrightarrow{x_i\ x_{\text{target}}}$ is larger than it.

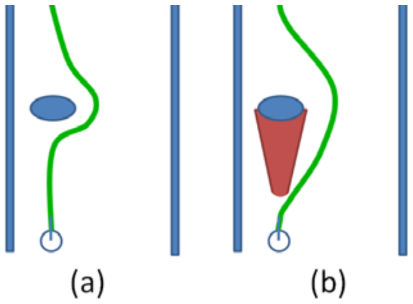


Fig. 8. Illustration of risk region

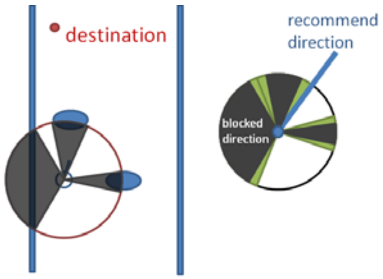


Fig. 9. Illustration of orientation decision

Furthermore, the RRT tends to generate path along the obstacle barrier, as shown in Figure 8(a). This is not a good property especially when the obstacle is moving. We set the risk region between obstacle and robot to make the planned path response to obstacle earlier. Those points in the risk region will have larger cost when connecting with the tree nodes to reduce the probability of planed path crossing through it.

- **Orientation Decision**

Although the RRT algorithm could be very fast in local-scale path planning, there are still some cases where the planning cannot be completed in real-time. Many works proposed the "anytime" version of RRT to cope with the time-limited problem. Although speeded up, still not guaranteed in real-time. In our system, we do not try to guarantee real-time generating the path but to find the appropriate direction in real-time for robot to follow. Because of the advantage of laser range finder, we can easily get the block distance in every angle degree. If the distance is smaller than a threshold, we determine that this angle degree is blocked, as shown in Figure 9. We also consider the angle degrees near the blocked angle degrees are in risk (green regions). Then, we choose the direction closest to the destination direction from the free degrees as the recommended direction. Once the RRT planning cannot be completed in deliberated time, we use the recommended direction for improving obstacle avoidance.

6 Experimental Results

6.1 Simultaneous Localization and Mapping

In the SLAM experiment, the robot is controlled to run a closed loop in a long corridor. For the outward part (downward), localization and building map at the same time. For the return part (turning & upward), localization is applied only. Figure 10 shows the results of SLAM using different landmarks. Table 1 shows the error between the starting point and ending point. Figure 10(a) shows the result using only odometer data. Figure 10(b) shows the result using the point landmarks. The result using the line landmarks is shown in Figure 10(c). And Figure 10(d) shows the result using both point and line landmarks.

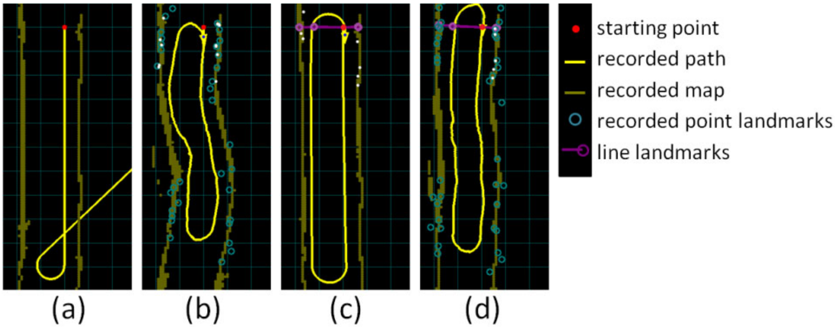


Fig. 10. Results of SLAM using different landmarks

Table 1. Error comparison

	X	Y
Ground Truth	0.0 m	0.0 m
(a)	6.23 m	-3.11 m
(b)	0.07 m	-0.41 m
(c)	0.13 m	-0.31 m
(d)	0.05m	0.00 m

From Figure 10 and Table 1, we can find that the result (a) has large error, especially when the robot is turning, the robot completely missed its orientation. Because (a) does not use any landmarks for error correction, the error is continuously accumulated. Therefore, the final estimated location has extremely large error. Result (b) uses the point landmarks for error correction and performs much better than (a). However, when the robot goes through a section with fewer point landmarks, the error of orientation becomes large and makes the built map distorted. Result (c) using the line landmarks, which are used for orientation correction, is almost perfect in orientation estimation. However, because there is no x-y lacation compensation, the straigh path is longer than the ground truth. Result (d), which is the method used in our system, can both correct the x-y location and orientation. Although there are still tiny error occuring on the recorded path, the robot can continuously compensate errors and find the location by itself.

6.2 Navigation

In the experiment of robot navigation, we first simulate our local-scale path planning method to see if it can really generate a good path for static and dynamic obstacle avoidance. Figure 11 shows the simulation environment and Table 2 shows the statistical results. The black squares in Figure 11 represent the static obstacles. The blue circles represent the dynamic obstacles, their speed and moving direction is randomly generated. As shown in Table 2, the success probability with three dynamic obstacles is higher than 90%, and the success probability with orientation decision is higher than that without orientation decision.

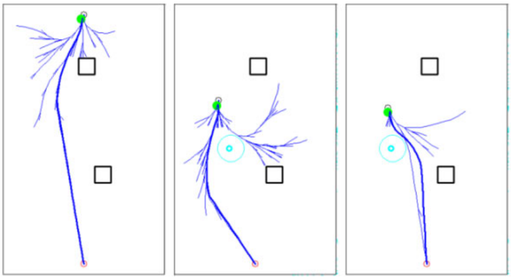


Fig. 11. Simulations of obstacle avoidance

Table 2. Success rate comparison

static obstacle number	dynamic obstacle number	Success(/100 times) (with orientation decision)	Success(/100 times) (without orientation decision)
2	1	99%	99%
2	2	96%	94%
2	3	91%	89%

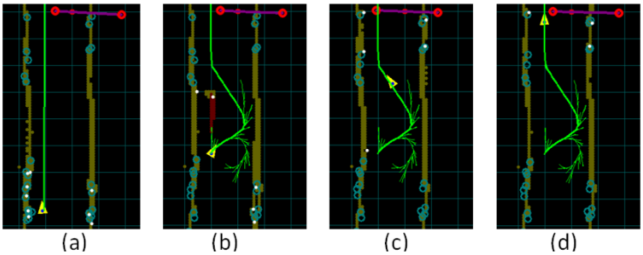


Fig. 12. On-road testing of obstacle avoidance

Beside the simulation results, we also have the on-road testing as shown in Figure 12. In Figure 12(a), the robot follows the original path planned by large-scale planning. The robot detects the obstacle and generates a substitute path for collision avoidance, as shown in (b). This on-road testing shows the robot can continuously localize itself, even if the obstacle hides some landmarks, and follow the substitute path to move to the destination.

7 Conclusion

In this paper, we propose a system using omni-directional camera and laser range finder for robot SLAM and navigation in indoor environments. We extract the point features and the line features as the landmarks. In SLAM and self-localization while navigation, the uncertainty of the odometer and observation is compensated by applying the linearized system model and odometer data to the extended Kalman filter (EKF). By the error compensation, the robot pose and the landmark feature can be well estimated. After the map has been built, robot can navigate in the environment based on it. We apply two-scale path-planning for navigation. The large-scale planning finds an appropriate path from starting point to destination. The A* with big grids is used in this step. The local-scale path-planning fills up the drawbacks of the prior step, such as dealing with the static and dynamic obstacles and smoothing the path for easier robot following. We apply an improved RRT algorithm for the path-planning in this step and use an orientation decision method to guarantee the real-time response to the detected obstacles.

Through the experiment results, we showed that the proposed system can smoothly and correctly locate itself, build the environment map and navigate in indoor environment. With the advantage of wide sensing field sensors, the self-localization still works even when there are obstacles covering some landmarks or the robot continuously changes the orientation to avoid collision.

References

1. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. In: Cox, I.J., Wilfong, G.T. (eds.) Autonomous Robot Vehicles, pp. 167–193. Springer, Heidelberg (1990)

2. Smith, R.C., Cheeseman, P.: On the representation and estimation of spatial uncertainty. Technical Report TR 4760 & 7239, SRI (1985)
3. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robotics & Automation Magazine 13(2), 99–110 (2006)
4. Bailey, T., Durrant-Whyte, H.: Simultaneous localization and mapping: part II. IEEE Robotics & Automation Magazine 13(3), 108–117 (2006)
5. Doucet, A., de Freitas, J.F.G., Murphy, K., Russel, S.: Rao-Blackwellized particle filtering for dynamic Bayesian networks. In: Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI), Stanford, CA, USA, pp. 176–183 (2000)
6. Civera, J., Davison, A.J., Montiel, J.: Inverse Depth Parametrization for MonocularSLAM. IEEE Trans. on Robotics 24(5), 932–945 (2008)
7. Chang, H.H., Lin, S.Y., Chen, Y.C.: SLAM for Indoor Environment Using Stereo Vision. In: Second WRI Global Congress on Intelligent Systems (2010)
8. Kuo, B.W., Chang, H.H., Lin, S.Y., Chen, Y.C., Huang, S.Y.: A Light-and-Fast SLAM Algorithm for Robots in Indoor Environments using Line Segment Map. Journal of Robotics (2011)
9. Dijkstra, E.W.: A note on two problems in connection with graphs. Numerische Mathematik 1, 269–271 (1959)
10. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics 4(2), 100–107 (1968)
11. Stentz: Optimal and Efficient Path Planning for Partially-Known Environments. In: Proceedings of 1994 IEEE International Conference on Robotics and Automation, vol. 4, pp. 3310–3317 (May 1994)
12. Zilberstein, S.: Using Anytime Algorithms in Intelligent Systems. AI Magazine (Fall 1996)
13. Likhachev, M., Ferguson, D., Gordon, G., Stentz, A., Thrun, S.: Anytime Dynamic A*: An Anytime, Replanning Algorithm. In: International Conference on Automated Planning & Scheduling (2005)
14. Overmars, M.: A random approach to motion planning. Tech. rep., Utrecht University (October 1992)
15. LaValle, S.M.: Rapidly-Exploring Random Trees: A New Tool for Path Planning. Tech. Rep. 98-11, Iowa State University, Ames, IA (October 1998)
16. Ferguson, D., Stentz, A.: Anytime RRTs. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, IROS (2006)
17. Ferguson, D., Stentz, A.: Anytime, Dynamic Planning in High-dimensional Search Spaces. In: Proceedings of the IEEE International Conference on Robotics and Automation (2007)

Color Based Stool Region Detection in Colonoscopy Videos for Quality Measurements

Jayantha Muthukudage[1], JungHwan Oh[1],
Wallapak Tavanapong[2], Johnny Wong[2], and Piet C. de Groen[3]

[1] Department of Computer Science and Engineering,
University of North Texas, Denton, TX 76203
[2] Computer Science Department, Iowa State University, Ames, IA 50011
[3] Mayo Clinic College of Medicine, Rochester, MN 55905
MuthukudageKumara@my.unt.edu, Junghwan.Oh@unt.edu,
{tavanapo,wong}@cs.iastate.edu

Abstract. Colonoscopy is the accepted screening method for detecting colorectal cancer or colorectal polyps. One of the main factors affecting the diagnostic accuracy of colonoscopy is the quality of bowel preparation. Despite a large body of published data on methods that could optimize cleansing, a substantial level of inadequate cleansing occurs in 10% to 75% of patients in randomized controlled trials. In this paper, we propose a novel approach that automatically determines percentages of stool areas in images of digitized colonoscopy video files, and automatically computes an estimate of the BBPS (Boston Bowel Preparation Scale) score based on the percentages of stool areas. It involves the classification of image pixels based on their color features using a new method of planes on RGB (Red, Green and Blue) color space. Our experiments show that the proposed stool classification method is sound and very suitable for colonoscopy video analysis where variation of color features is considerably high.

Keywords: Image Classification, Region of Interest Detection, Colonoscopy, Medical Image Analysis.

1 Introduction

Advances in video technology are being incorporated into today's healthcare practices. Various types of endoscopes are used for colonoscopy, upper gastrointestinal endoscopy, enteroscopy, bronchoscopy, cystoscopy, laparoscopy, wireless capsule endoscopy, and some minimally invasive surgeries (i.e., video endoscopic neurosurgery). These endoscopes come in various sizes, but all have a tiny video camera at the tip of the endoscope. During an endoscopic procedure, this tiny video camera generates a video signal of the interior of a human organ, which is displayed on a monitor for real time analysis by the physician. Colonoscopy is an important screening tool for colorectal cancer. In the US, colorectal cancer is the second leading cause of all cancer deaths behind lung cancer [1]. As the name implies, colorectal cancers are malignant tumors that develop in the colon and rectum. The survival rate

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 61–72, 2011.

is higher if the cancer is found and treated early before metastasis to lymph nodes or other organs occurs.

The effectiveness of colonoscopy in prevention of colorectal cancers depends on the quality of the inspection of the colon, which generally can be evaluated in terms of the withdrawal time (time spent during the withdrawal phase) and the thoroughness of the inspection of the colon mucosa. Current American Society for Gastrointestinal Endoscopy (ASGE) guidelines suggest that (1) on average the withdrawal phase during a screening colonoscopy should last a minimum of 6 minutes and (2) the visualization of cecum anatomical landmarks such as the appendiceal orifice and the ileocecal valve should be documented [2].

Nevertheless, there was no automated measurement method to evaluate the endoscopist's skill and the quality of a colonoscopic procedure. To address this critical need, we have developed a prototype capturing system, which automatically records colonoscopic procedures on a hard disk in MPEG-2 format [3]. This system has been placed at Mayo Clinic Rochester since the beginning of February 2003 to capture colonoscopic procedures performed by de Groen (co-author) and colleagues.

The diagnostic accuracy of colonoscopy depends on the quality of the bowel preparation [4]. Inadequate cleansing can result in missed pathologic lesions. Colonic cleansing is mostly performed with solutions containing polyethylene glycol (PEG), and the alternatives are sodium phosphate, magnesium citrate, or bisacodyl [5]. The ideal preparation method would reliably empty the colon of all fecal material, and have little effect on the gross or the microscopic appearance of the mucosa. It would require a relatively short period for ingestion and evacuation, cause little patient discomfort, and produce no significant fluid-electrolyte shifts. It also should maximize the detection of colonic disease including polyps and carcinoma [5].

The quality of bowel cleansing is generally assessed by the quantity of solid or liquid stool in the lumen. Despite a large body of published data on methods that could optimize cleansing, a substantial level of inadequate cleansing occurs in 10% to 75% of patients in randomized controlled trials [6]. Poor bowel preparation has been associated with patient characteristics, such as inpatient status, history of constipation, use of antidepressants, and noncompliance with cleansing instructions. The American Society for Gastrointestinal Endoscopy (ASGE) and American College of Gastroenterology (ACG) Taskforce on Quality in Endoscopy suggested that every colonoscopy report should include an assessment of the quality of bowel preparation. They proposed the use of terms such as “excellent,” “good,” “fair,” and “poor,” but admitted that these terms lack standardized definitions [7]. To address this, the authors in [7] proposed the ‘Boston Bowel Preparation Scale’ (BBPS) in which the terms “excellent,” “good,” “fair,” and “poor,” were replaced by a four-point scoring system applied to each of the three broad regions of the colon: the right colon (including the cecum and ascending colon), the transverse colon (including the hepatic and splenic flexures), and the left colon (including the descending colon, sigmoid colon, and rectum). This scoring system will be discussed in Section 4 later.

This method is still based on some subjective evaluation of colon parts (i.e., just numbers instead of terms). An automatic method to identify stool in digitized images obtained during colonoscopy would obviate any subjective scoring methods and be a valuable asset among automated tools for measuring the quality of colonoscopy procedures.

A frame in colonoscopy video consists of a number of pixels as other digitized images typically do. Each pixel has three number values representing Red (R), Green (G), and Blue (B), so each pixel can be plotted into 3-dimensional RGB color space. A set of stool pixels can form arbitrary shape(s) of volume(s) in 3-D RGB color space. If we can represent the volume(s) mathematically, we can decide automatically whether a pixel is a stool pixel or not. For the mathematical representation, we propose to use a set of planes (which will be discussed in more detail in Section 3). In this study, we propose two methods as follows:

- a method automatically deciding a percentage of stool area for each frame of colonoscopy video, and
- a method automatically computing an estimate of BBPS score based on the percentages of stool areas.

The rest of the paper is organized as follows. Section 2 discusses the background and related work. Section 3 and Section 4 describe the proposed methodology and the calculation of BBPS score, respectively. Section 5 shows our experimental results. Finally, Section 6 summarizes our concluding remarks.

2 Background and Related Work

Much research has focused on color based classification of images. In fact, color based classification plays a major role in the field of medical imaging. A huge number of articles based on color features has been published. The most common problem discussed within these articles deals with new positive class examples emerging after the training processes finish. We will briefly discuss two examples.

Zeki et al [8] proposed an incremental learning algorithm with SVM (Support Vector Machine) ensembles. The authors mainly focus on how to overcome a catastrophic misclassification problem of the SVM classifier by adding the ability to learn new instances. They propose to create a new ensemble of SVM classifier for the newly added data instances. The authors suggested to generate a number of classifiers for a given data set, and to keep a subset of most effective classifiers that are selected based on a weighted majority voting system. They combined the Learn++ algorithm with SVM to give it the ability to learn new instances. A major shortcoming of this approach is that the method is not applicable if there is not enough new data available to train a new SVM ensemble.

In our previous work [9], we proposed a method classifying stool images in colonoscopy videos using a SVM classifier. For each frame a vector is specified, and a color histogram is computed for each frame. The video frame is down-sampled into blocks in order to reduce the size of the feature vector. Features to the SVM classifier are, in fact, the mean value for each block. Then, the stool mask is applied to each video frame using the trained SVM classifier, and a post processing step is applied to improve the detection quality. The post processing step includes a majority filter and a binary area opening filter. Finally, frames having more than 5% of stool area are classified as stool frames. It also has the catastrophic misclassification problem when it comes to learn new instances. And it lacks the ability to learn new data instances when available.

Our new method is preferable in detecting stool regions over the above mentioned methods because of the following;

- Our new method addresses the catastrophic misclassification problem (incremental learning) of SVM classifier. It can learn new instances instantly, and does not need a certain amount of data as in SVM classifier. Consequently, it is more accurate.
- Since our method can learn new instances instantly, its training process is fast. Also, its detecting process is fast because we can optimally reduce the number of comparisons.

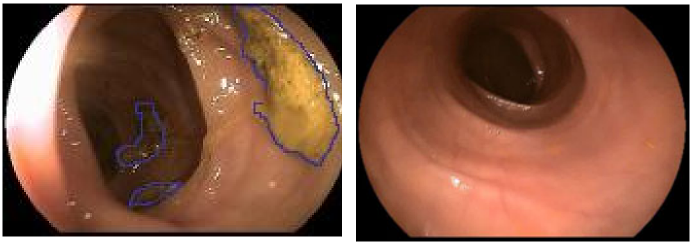
3 Classification and Detection Methodology

The proposed method has the training and the detecting (or test) stages. The training stage has three steps: All stool pixel projection, Stool plane selection, and Stool plane modeling. As a result, the training stage generates a classification model which is used in the detecting stage. In this section, we will discuss the two stages.

3.1 Training Stage

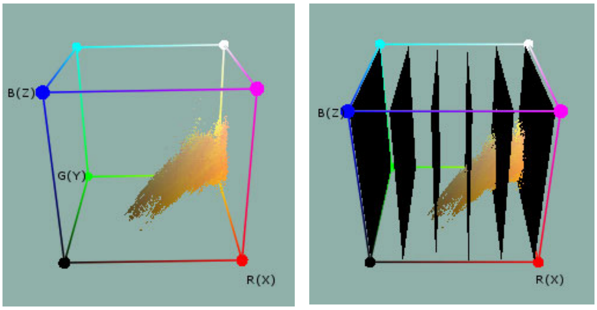
Digital color images including our colonoscopy images are modeled in a RGB color space (cube) in which each color band is represented with 8-bit ranging from 0 to 255, giving us a total of 256^3 potential colors. Fig. 1 shows an example of frames that can be found in a colonoscopy video. We project all stool pixels in a frame into RGB color cube as the first step (All stool pixel projection). To discriminate stool pixels from non-stool pixels we use the fact that each color pixel has a unique location in the RGB color cube as three coordinates R, G, B as illustrated in Fig. 2(a). For convenience, RGB is mapped to the XYZ coordinate system as shown. In the second step (Stool plane selection), we put 256 planes into the RGB cube along the R (X) axis so that each integer location on the R axis has a plane parallel to a GB plane as seen in Fig. 2 (b). It is possible to put planes along the G or B axis – doing so will not alter the classification modeling. In our study we selected the R axis along which we put planes. We assign a number (from 0 to 255) to each plane (i.e., Plane#0, Plane#1, … Plane#255). One assigned number to each plane is sufficient since all planes are perpendicular to the GB (YZ) plane. Among these 256 planes, we select only planes with stool pixels. Each selected plane is called a ‘Positive Plane’.

Each positive plane contains a projection of stool pixels at the corresponding location, and is treated as a 2D classifier at the relevant location. For instance, Plane#0 at the location (0, 0, 0) is treated as a classifier for positive class examples (stool pixels in our case) that has a R (X) value of zero (0). This method inherits fast classification as it already possesses the property of eliminating non-relevant class examples (i.e., non-stool pixels in our case) in the training process.



(a) (b)

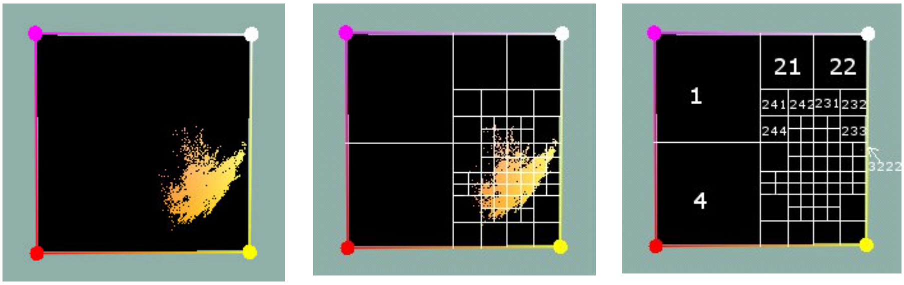
Fig. 1. Examples of (a) Stool- marked with blue (b) Non-stool Frames



(a) (b)

Fig. 2. (a) RGB cube and corresponding locations of stool pixels, and mapping of RGB axis to XYZ axis (within brackets), and (b) Several planes inserted into the RGB cube of (a)

In the third step (Stool plane modeling), we model the areas of positive class examples (stool pixels). First, a positive plane (Fig. 3(a)) is divided into four blocks. A block is a square since each plane is a square (256 x 256), and each may contain all stool pixels, all non-stool pixels, or mixture of non-stool and stool pixels. For all four blocks, we check the following three conditions. If all (or more than 95%) of pixels in a block are positive class examples (stool pixels), the block becomes a positive block, and the procedure for this block is done. If all pixels in a block are non-positive class examples (non-stool pixels), then the block becomes a negative block, and the procedure for this block is done. If some (less than 95%) of pixels in a block are positive class examples (stool pixels), and the block has more than or equal to the MNP (Minimum Number of Pixels – MNP is 16 in our case), the block is divided into four smaller blocks, and we check the above three conditions for all four smaller blocks. The minimum block size is 4 x 4. When the iteration reaches the minimum block size, a block becomes a positive block if it has more positive class examples (stool pixels). Otherwise, it becomes negative block. This procedure is recursive, and the blocks become smaller in the next iteration (Each block is divided by four at each iteration). In case the block has less than the MNP and it has more positive class examples (stool pixels), then it becomes a positive block. Otherwise, it becomes a negative block. All levels of blocks have their own unique number values in the way shown in Fig. 3(c). It is a very convenient and non-ambiguous way for numbering. Among these numbers, a set of numbers for positive blocks can form a vector for a positive plane, and a set of vectors from all positive planes can form a classification model for the detecting stage. This model possesses an incremental learning property. Its incremental learning is performed as follows. When there is a new positive pixel to be inserted into the model, we can find a corresponding minimum size (4 x 4) block which is a negative block. The block can become a positive block if it gets more positive class examples (stool pixels) by adding this new positive pixel. In this way, we do not have to run the entire training process from the beginning when we need to add additional positive examples.



(a) (b) (c)

Fig. 3. (a) Positive class examples (stool pixels) projected on a positive plane (plane#250) as looking into the RGB cube from right side in Fig. 2(b), (b) Minimum coverage area of the positive classes examples (stool pixels), and (c) Unique numbering of blocks for fast access (not all shown for clarity) – clockwise numbering starting from top left quadrant

3.2 Detecting Stage

Detection of the stool pixel is performed by evaluating a candidate pixel on the classification model generated in Section 3.1. Once there is pixel to be detected, the R (X) value of the pixel is obtained and used as the index to select the corresponding positive plane. For example, if the R (X) value is 5, then Plane#5 is selected and examined. This will dramatically reduce the number of comparisons so that the analysis time is significantly reduced. In other words, the analysis time to determine pixel stool class of the proposed technique is not dependent on the number of positive planes, but on how many positive blocks there are in the corresponding plane, which is usually very small. By comparing the GB (YZ) values of the pixel with the vector obtained from the third step of the training stage (discussed in Section 3.1), we can determine whether it can be classified as a positive class (stool) pixel. Otherwise, it is classified as negative (non-stool) class pixel. After all pixels of a frame are evaluated, we can calculate a percentage of stool area for each frame: the number of all stool pixels divided by the number of total pixels.

4 Computing BBPS Score

In this section, we will discuss a method automatically compute an estimate of the BBPS score based on the percentages of stool areas obtained in Section 3. As mentioned in Section 1 (Introduction), the authors in [7] proposed ‘Boston Bowel Preparation Scale’ (BBPS) in which the terms “excellent,” “good,” “fair,” and “poor,” were replaced by a four-point scoring system applied to each of the three broad regions of the colon: the right colon (including the cecum and ascending colon), the transverse colon (including the hepatic and splenic flexures), and the left colon (including the descending colon, sigmoid colon, and rectum). These six parts of colon can be seen in Fig. 4, and the relationships between the terms and the points can also be seen in Table 1.

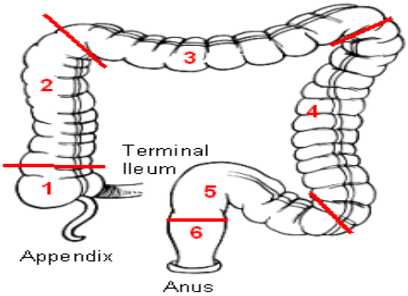


Fig. 4. Six parts of Colon: 1 - Cecum, 2 - Ascending colon, 3 - Transverse colon, 4 - Descending colon, 5 - Sigmoid, and 6 - Rectum

Table 1. Relationship between the quality terms and the quality points

Quality Term	Quality Point
Excellent	3
Good	2
Fair	1
poor	0

The points in Table 1 are assigned as follows:

- 0 = Unprepared colon segment with mucosa not seen due to solid stool that cannot be cleared.
- 1 = Portion of mucosa of the colon segment seen, but other areas of the colon segment not well seen due to staining, residual stool and/or opaque liquid.
- 2 = Minor amount of residual staining, small fragments of stool and/or opaque liquid, but mucosa of colon segment seen well.
- 3 = Entire mucosa of colon segment seen well with no residual staining, small fragments of stool or opaque liquid.

Each region of the colon receives a “segment score” from 0 to 3 and these segment scores are summed for a total BBPS score ranging from 0 to 9. Therefore, the maximum BBPS score for a perfectly clean colon without any residual liquid is 9, and the minimum BBPS score for an unprepared colon is 0.

We compute an estimate of the BBPS score automatically for a recorded colonoscopy video. A colonoscopy video consists of two phases: an *insertion phase* and a *withdrawal phase* as seen in Figure 5. During the insertion phase, a flexible endoscope (a flexible tube with a tiny video camera at the tip) is advanced under direct vision via the anus into the rectum and then gradually into the cecum (the most proximal part of the colon) or the terminal ileum. During the withdrawal phase, the endoscope is gradually withdrawn. The purpose of the insertion phase is to reach the cecum or the terminal ileum. Careful mucosa inspection and diagnostic or therapeutic interventions such as biopsy, polyp removal, etc., are performed during the withdrawal phase.

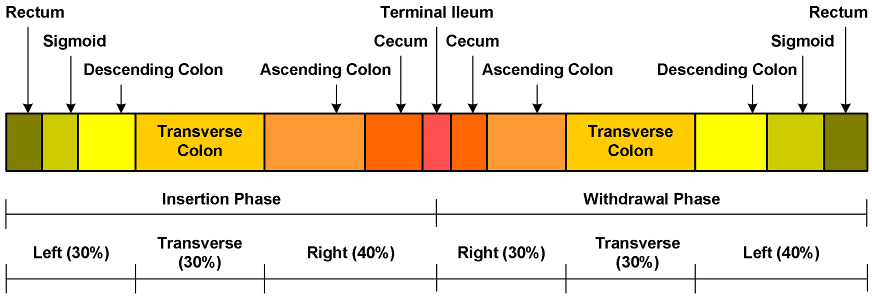


Fig. 5. Two Phases in Colonoscopy Video

The recorded colonoscopy video is divided into insertion phase and withdrawal phase automatically using the techniques we developed [10]. In our estimate of BBPS implementation, the right colon has the last 40% of insertion phase plus the first 30% of withdrawal phase. The transverse colon has the middle 30% of insertion phase plus the middle 30% of withdrawal phase. The left colon has the first 30% of insertion phase plus the last 40% of withdrawal phase. These numbers are based on experiments and opinion of the domain expert. We calculate estimated BBPS score values mathematically based on the stool percentage values obtained above for each frame. We assign a score value for each frame based on the stool pixel percentage present in the frame, and calculate the numerical average for each colon segment (right colon, transverse colon, and left colon) for the final score value. The stool percentage values and the corresponding score values are estimates based on the original images of the BBPS description and are shown in the table 2.

Table 2. Stool percentage in a frame and the assigned score value

Stool percentage %	Score value assigned
0 – 10	3
11- 25	2
26 – 50	1
51 – 100	0

5 Experimental Results

All the computations in our experiments were performed on a PC-compatible workstation with an Intel Pentium D CPU, 1GB RAM, and Windows XP operating system. For our experiments, we used 58 videos recorded with Fujinon colonoscopes. The average length of the videos is around 20 minutes, and their frame size is 720 x 480 pixels. This section is divided into two subsections; one for assessing the proposed stool detection and the other for BBPS score calculation.

5.1 Stool Detection

We randomly extracted 1,000 frames from all 58 videos, in which each frame has at least one stool region. The domain experts marked and confirmed the positive (stool) regions in these frames. From half (500) of these frames, we filtered out duplicate examples (pixels), and obtained only unique positive examples (stool pixels) for the training. Table 3 shows the stool and non-stool pixels used for training. Using 31,858 stool pixels, we followed all the steps in Section 3.1. Then, we used all the pixels in the remaining half (500) of the frames for the detecting stage discussed in Section 3.2. We assess the effectiveness of our proposed algorithm at the pixel level by determining the performance metrics *Sensitivity* and *Specificity*. For a comparison purpose, we implemented the method in our previous work [9] using the same dataset mentioned above (also seen in Table 3). Table 4 shows this comparison. As seen in the table, the proposed method is better than the previous one in terms of sensitivity and specificity.

Table 3. Number of examples (pixels) used in the training stage

	Stool Dataset
Positive (stool)	31,858
Negative (non-stool)	52,434

Table 4. Performance comparison with previous work

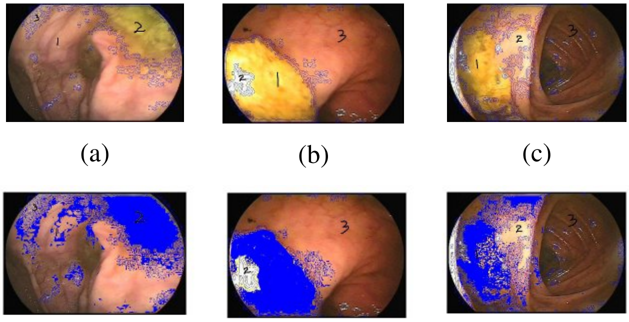
Sensitivity		**Specificity**	
New	Old	New	Old
92.9 (%)	90.6	95.0	93.8

Also, we implemented the well-known KNN (K-Nearest Neighbor, K=1 in our study) classifier using the same dataset mentioned above to see how fast the proposed method can perform. Table 5 presents the speed comparison for KNN classifier with our proposed method. It takes more than 420 seconds (7 minutes) to evaluate a frame (720 x 480 pixels) in the KNN. On average, it takes 0.00127 seconds to evaluate one pixel. However, it takes around 11 seconds to evaluate a frame (720 x 480 pixels) in the proposed method. If we consider only the detection stage, it takes less than one second to evaluate one frame. This is a significant achievement. We need to process 3,600 frames to generate a colonoscopy report for a 20 minute colonoscopy video if we analyze 3 frames per second (3 frames/second * 60 seconds/minute * 20 minutes = 3600 frames). Thus, it is not practical to use KNN classifier even though it can provide 98% of sensitivity and specificity on average.

Table 5. Average Time taken for KNN and the proposed method

KNN (trainging +detection)	Proposed method (Detection)	Prposed Method (Training)
437.5 (seconds)	0.9	10.0

Fig. 6 lists some results obtained using the proposed method. The numbers (1, 2 and 3) on each frame represent the regions semi-automatically segmented for the determination of ground truth. For instance, region 2 in Fig. 6(a), region 1 in Fig. 6(b), and region 1 in Fig. 6(c) were labeled as stool by the domain experts. The first row consists of the original frames with the ground truth marked, and the second row contains the results from our method for the first row (stool regions are marked with blue).



(a) (b) (c)

Fig. 6. Sample results for stool detection

5.2 BBPS Calculation

We generated estimates of BBPS scores for all 58 videos and list randomly selected 5 results in Table 6 having a comparison with the ground truth scores suggested by domain experts. The column 'Ground truth BBPS' in Table 6 is the average score values from three different experts. It is rare to find video files where all three experts agree this close about the scores. Therefore, it is hard to find a definitive Ground Truth score for a given video. We took the average of three BBPS ground truths as our target value to be reached. As seen in the table, the calculated values are very close to the ground truths.

Table 6. Comparision of Calculated BBPS scores with Ground Truth BBPS scores

Video ID	Calculated BBPS	Ground Truth BBPS
1.mpg	7	6
5.mpg	4	3
9.mpg	6	6
10.mpg	6	7
13.mpg	6	5

6 Conclusion and Future Work

The two most critical aspects of colonoscopy that determine its protective effect against development of colorectal cancer are the quality of bowel cleansing and technical performance of the endoscopist. Neither of these two aspects can be objectively measured in a manual fashion. Recent reports of "missed" polyps and cancers suggest that the protective effect of colonoscopy is far from complete, raise questions about the quality of bowel cleansing as well as technical performance, and call for new methods to measure and improve the quality of colonoscopy. Indeed, automated video analysis techniques have recently been introduced to objectively determine technical performance. In this paper, we present an automated method to detect stool regions based on the color features using new classification modeling. Our preliminary investigation shows that our stool detection method is able to detect stool with very high accuracy achieving sensitivity over 93% with 95% specificity. Our previous work for stool detection [9] was very good, but had its limitations. First, the training data were images derived from a single patient; but in this new study, we used 58 different videos from 58 different patients. Second, a global, objective "colon cleansing" score was needed to be developed representing a composite of all individual image scores and tested against one or more manual "colon cleansing" scores. In this new study, we implemented a method to compute an estimate of the BBPS score automatically, and compared the score with the ground truths provided by domain experts. Our new method shows improved performance and can be applied in colonoscopy practice for quality measurements. In addition, our method has the ability to learn new positive class examples without running the entire training process from beginning as we can adjust each plane separately. This adds to our method the valuable ability of incremental learning. However, further research is necessary. For example, stool varies in consistency from solid lumps to transparent water-diluted fluid; our training data consisted of images where the outline of solid or non-transparent, liquid stool was marked as "stool". Thus mucosa with just a thin cover of semi-transparent liquid stool was not included in our training data. This means that not the entire amount of stool was targeted for recognition. Clearly, more testing is needed to determine how well our algorithms hold up under variable, real-life circumstances.

Acknowledgement. This work is partially supported by NSF STTR-Grant No. 0740596, 0956847, National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK DK083745), and the Mayo Clinic. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of authors. They do not necessarily reflect the views of the funding agencies. Johnny Wong, Wallapak Tavanapong, and JungHwan Oh hold positions at EndoMetric Corporation, Ames, IA 50014, U.S.A, a for profit company that markets endoscopy-related software. Johnny Wong, Wallapak Tavanapong, JungHwan Oh, and Mayo Clinic own stocks in EndoMetric. Piet de Groen, Johnny Wong, Wallapak Tavanapong, JungHwan Oh and Mayo Clinic have received royalty payments from EndoMetric.

References

1. American Cancer Society.:Colorectal Cancer Facts and Figures, American Cancer Society Special Edition 2005, pp. 1–20 (2005)
2. Douglas, K.R., John, L.P., Todd, H.B., Amitabh, C., Jonathan, C., Stephen, E.D., Brenda, H., Brian, C.J., Klaus, M., Bret, T.P., Michael, A.S., Douglas, O.F., Irving, M.P.: Quality Indicators for Colonoscopy. American Journal of Gastroenterology 101, 873–885 (2006)
3. Stanek, S., Tavanapong, W., Wong, J., Oh, J., de Groen, P.C.: Automatic Real-Time Capture and Segmentation of Endoscopy Video. PACS and Imaging Informatics. In: SPIE Medical Imaging, vol. 6919, pp. 69190X-69190X-10 (February 2008)
4. Cappel, M.S., Friedel, D.: The Role of Sigmoidoscopy and Colonoscopy in the Diagnosis and Management of Lower Gastrointestinal Disorders: Endoscopic Findings, Therapy, and Complications. Medical Clinics of North America 86, 1253–1288 (2002)
5. Ernstoff, J.J., Howard, D.A., Marshall, J.B., Jumshyd, A., Mc-Cullough, A.J.: A Randomised Blinded Clinical Trial of a Rapid Colonic Lavage Solution (Golytely) Compared with a Standard Preparation for Colonoscopy and Barium Enema. Gastroenterology 84, 1512–1516 (1983)
6. Ness, R.M., Manam, R., Hoen, H.J., Chalasani, N.: Predictors of Inadequate Bowel Preparation for Colonoscopy. American Journal of Gastroenterology 96, 1797–1802 (2001)
7. Lai, E.J., Calderwood, A.H., Doros, G., et al.: The Boston Bowel Preparation Scale: a valid and reliable instrument for colonoscopy-oriented research. Gastrointestinal Endoscopy Clinics of North America 69, 620–625 (2009)
8. Erdem, Z., Polikar, R., Gurgen, F., Yumusak, N.: Ensemble of SVMs for Incremental Learning. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 246–256. Springer, Heidelberg (2005), doi:10.1007/11494683_25
9. Hwang, S., Oh, J., Tavanapong, W., Wong, J., de Groen, P.C.: Stool Detection in Colonoscopy Videos. In: Proc. of International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Vancouver, British Columbia, Canada, pp. 3004–3007 (August 2008)
10. Oh, J., Rajbal, M.A., Muthukudage, J.K., Tavanapong, W., Wong, J., de Groen, P.C.: Real-Time Phase Boundary Detection in Colonoscopy Videos. In: Proc. of 6th International Symposium on Image and Signal Processing and Analysis (ISPA 2009), Salzburg, Austria, September 16-18, pp. 724–729 (2009)

Improving Motion Estimation Using Image-Driven Functions and Hybrid Scheme

Duc Dung Nguyen and Jae Wook Jeon*

Department of Electrical and Computer Engineering, Sungkyunkwan University, Korea
nddunga3@skku.edu, jwjeon@yurim.skku.ac.kr

Abstract. We introduce an alternative method to improve optical flow estimation using image data for control functions. Base on the nature of object motion, we tune the energy minimization process with an image-adaptive scheme embedded inside the energy function. We propose a hybrid scheme to improve the quality of the flow field and we use it along with the multiscale approach to deal with large motion in the sequence. The proposed hybrid scheme take advantages from multigrid solver and the pyramid model. Our proposed method yields good estimation results and it shows the potential to improve the performance of a given model. It can be applied to other advanced models. By improving quality of motion estimation, various applications in intelligent systems are available such as gesture recognition, video analysis, motion segmentation, etc.

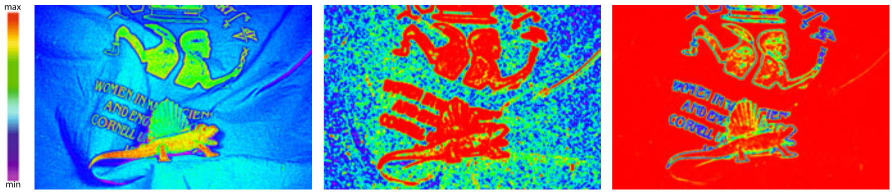
1 Introduction

Motion estimation is still an active field in computer vision with various applications, including motion segmentation, video understanding, and gesture recognition. Optical flow, in particular, has been developed and improved in various ways for almost three decades. Several models and techniques have been proposed to enhance the quality of optical flow, since the first approach of Horn and Schunck [1] and Lucas-Kanade [2]. It is important to detect the object motion rather than pixel-wise intensity matching (e.g. optical flow) for real-life applications. Thus, the occlusion problem in optical flow estimation must be taken into account. The sharpness of the flow field along the object boundary is also important in the motion segmentation task.

The intensity difference constraint and the smoothness constraint, which are well-known in the literature, do not exactly describe the object movement in a real scene. In [3], the author tends to minimize the energy function with both the intensity constraint and smoothness constraint embedded inside. This model may hold in normal circumstances but not in the case of occlusion. In such a case, the intensity constraint does not hold. The flow field might be shifted to somewhere else but not the occluded area due to the energy function minimization. Thus, the partial differential equation (PDE)

* This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-(C1090-1121-0008)), and by Priority Research Centers Program through the National Research Foundation of Korea (NRK) funded by the Ministry of Education, Science and Technology (2011-0018397).

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 73–84, 2011.



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

Fig. 1. Visual result of adapting function on Dimetrodon sequence [10] (a) the color code for image and control functions, (b) enhanced image of frame 10, (c) data adapting-function f_d, (d) smoothness adapting-function

is no longer a favored tool to solve this problem as better solvers are available. In [4], the authors also use the variational model similar to [3] to initialize the flow field. They employ the color segmentation with flow field information to improve the estimation. Recently, [5] reveals excellent results performed by the variational model with the help of color information and improvements in the regularizers. Among the best, the total variational methods are also very strong solutions for this problem, as [6,7,8,9] yields the top results on Middlebury's website. So far, many improvements have been made to enhance the estimation result of optical flow. Yet, we can still push quality of optical flow estimation further. The key answer for this lies in the nature of object motion and the purpose of the estimation model. We will show how to improve a given model by using advanced scheme.

In this paper, we propose a model that can adapt the estimation process using the image information. We start from basic constraints of optical flow and use PDE solver for energy minimization to prove that our proposed method can improve the quality of the flow field. We propose the hybrid solver, which takes advantage of the multi-grid solver and coarse-to-fine estimation scheme, to deal with large displacement. With some small adaptations, we can even speed up the estimation process. Given an estimation model, our proposed scheme can push the quality of the estimation result further. The adapting functions and the hybrid scheme are the keys in our method. In the next section, we will discuss the proposed model and how the image information can be embedded in the model. We introduce details the hybrid scheme we use to solve the energy minimization problem in Section 3. We will detail the implementation and experiments in Section 4. We summarize the paper and outline on future work in Section 5.

2 Image-Adapting Energy Function

2.1 Optical Flow Constraints

Intensity Constraint. This constraint is the most basic constraint in every optical flow estimation model. It can be stated as follows

$$\mathbf{p} = \mathbf{argmin} |I_{t+\Delta t}(\mathbf{x}+\mathbf{p}) - I_t(\mathbf{x})| \tag{1}$$

where $I_t(\mathbf{x}) : \mathbb{R}^2 \mapsto \mathbb{R}$ denotes the image intensity of point (x,y) at time t with $\mathbf{x} = (x,y)^T$, and $\mathbf{p} = (u,v)^T$ is the motion vector between an image at time t and another image at time $t+\Delta t$. This constraint is mostly described in the literature as the equation $I_{t+\Delta t}(\mathbf{x}+\mathbf{p})-I_t(\mathbf{x})=0$; by which the linearized form yields the well-known optical flow constraint [1]:

$$I_x u + I_y v + I_t = 0. \tag{2}$$

The gradient constraint, conversely, is less sensitive to slight changes in brightness. However, it only holds when an object undergoes translation motion but not in the general case. Therefore, we only use the intensity constraint in the model.

Smoothness Constraint. This constraint states that the motion field must be smooth inside the object, even the object undergoes complex motion. In addition, the aperture problem occurs when the gradient disappears, or when the flow can only be detected in normal direction to the gradient. This is solved by considering the flow field smoothness. The flow field discontinues along the object boundary to achieve optimal estimation. With $f_s(\mathbf{x})$ as a function of image data, we formally express this piecewise smoothness constraint as follows

$$\mathbf{p} = \mathbf{argmin}\left(f_s(\mathbf{x})\left(|\nabla u|^2 + |\nabla v|^2\right)\right). \tag{3}$$

2.2 Adapting Functions

We arrive at the energy function used by previous work [3,11,12] using the above constraints:

$$\begin{aligned} E &= E_{data} + \beta E_{gradient} + \alpha E_{smooth} \\ &= \int_\Omega \left[\varphi_d(\mathbf{x},\mathbf{p}) + \beta\varphi_g(\mathbf{x},\mathbf{p}) + \alpha\varphi_s(\mathbf{x},\mathbf{p})\right] d\mathbf{x} \end{aligned} \tag{4}$$

where $\varphi_d(\mathbf{x},\mathbf{p})$, $\varphi_g(\mathbf{x},\mathbf{p})$ and $\varphi_s(\mathbf{x},\mathbf{p})$ are data term, gradient term and smoothness term respectively. In this work, we drop the gradient term, as mentioned above, and inject the adapting functions in the energy functions as follows:

$$\varphi_d(\mathbf{x},\mathbf{p}) = \varphi_d\left(f_d(\mathbf{x})|I_{t+\Delta t}(\mathbf{x}+\mathbf{p}) - I_t(\mathbf{x})|^2\right) \tag{5}$$

$$\varphi_s(\mathbf{x},\mathbf{p}) = \varphi_s\left(f_s(\mathbf{x})\left(|\nabla u|^2 + |\nabla v|^2\right)\right) \tag{6}$$

where f_d, f_s are adapting functions that will tune the estimation process using image information. The image itself contains much information. The idea is that we can suppress the difference of the data and flow, based on the features of the current pixel, given image information. In this way, the model can adapt to various kinds of image sequences and yields better estimation results.

Data Adaptation. We design f_d to suppress the difference in intensity of the points inside the object. First, the flow field inside the object must be smooth, since the smoothness constraint will drive the flow inside the object. Conversely, we can deal with the occlusion problem simultaneously. When a part of the object is occluded in the next frame, this data term still holds if the occluded area is inside the object. Fig. 1 shows the visual view of control functions on the Dimetrodon sequence [10].

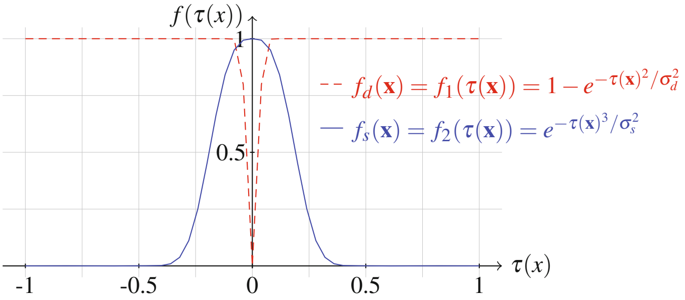


Fig. 2. Plot of control functions f_d(red dashed line) and f_s(blue line)

We can introduce several forms of f_d to yield the same effect based on the image features. Here, we introduce f_d as a function of gradient magnitude. Let $\tau(\mathbf{x}) = |\nabla I|$ be the magnitude of the image gradient at $\mathbf{x}$, then f_d can be simply defined as

$$f_d(\mathbf{x}) = f_1(\tau(\mathbf{x})) = 1 - e^{-\tau(\mathbf{x})^2/\sigma_d^2} \tag{7}$$

As in Fig. 2, the data difference is suppressed when the point is inside the homogenous area, e.g. $|\nabla I| \approx 0$. The data difference includes intensity difference, gradient difference, and other measurements, such as Hessian. In this work, we are concerned about the intensity difference in the model. Other measurements obviously can be controlled by this function, without a problem, since it is a function of spatial position. The parameter σ_d has an important role in the estimation result. We choose σ_d sufficiently small, so that it cannot create an over-smooth effect in the final result. Many experiments have been performed and we choose $\sigma_d = \sqrt{0.001}$ that yields the most stable results among test sequences.

Smoothness Adaptation. The function f_s should be large in homogenous area due to the smoothness energy (6), so that the flow field inside the object will be as smooth as possible. Similar to the data control function f_d above, f_s can be defined as follows

$$f_s(\mathbf{x}) = f_2(\tau(\mathbf{x})) = e^{-\tau(\mathbf{x})^\lambda/\sigma_s^2}. \tag{8}$$

where λ and σ_s are parameters controlling the shape of f_s. Setting λ to 2, we will get a similar form to f_d. However, this is not the case for f_s. The shape of function f_s must be wider and slowly drop, as in Fig. 2. When f_s drops too fast, the discontinuity of the flow will appear at some area where the gradient magnitude is larger than the specific threshold. This creates a segmentation effect on the flow field that we do not really want; especially, when the scene has smooth areas, where the gradient only changes a little bit from one to the next. This analysis leads to the f_s in (8) with $\lambda = 3$ and $\sigma_s = 0.1$. Other designs of f_d and f_s are available and can yield the same result, if they satisfy the above descriptions.

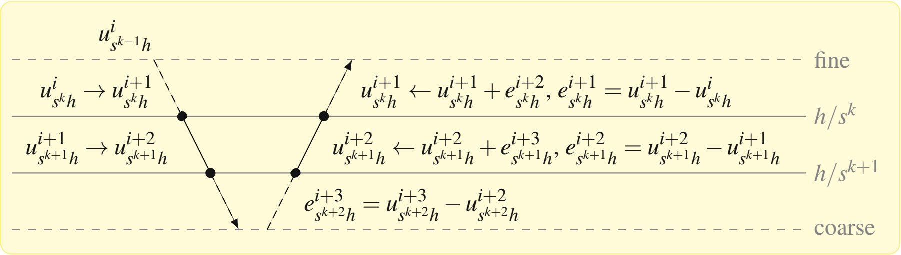

$u^i_{s^{k-1}h}$
fine
$u^i_{s^k h} \to u^{i+1}_{s^k h}$
$u^{i+1}_{s^k h} \leftarrow u^{i+1}_{s^k h} + e^{i+2}_{s^k h}$, $e^{i+1}_{s^k h} = u^{i+1}_{s^k h} - u^i_{s^k h}$
h/s^k
$u^{i+1}_{s^{k+1}h} \to u^{i+2}_{s^{k+1}h}$
$u^{i+2}_{s^{k+1}h} \leftarrow u^{i+2}_{s^{k+1}h} + e^{i+3}_{s^{k+1}h}$, $e^{i+2}_{s^{k+1}h} = u^{i+2}_{s^{k+1}h} - u^{i+1}_{s^{k+1}h}$
h/s^{k+1}
$e^{i+3}_{s^{k+2}h} = u^{i+3}_{s^{k+2}h} - u^{i+2}_{s^{k+2}h}$
coarse

Fig. 3. Hybrid scheme is used to solve Euler-Lagrange equations with scale parameter s

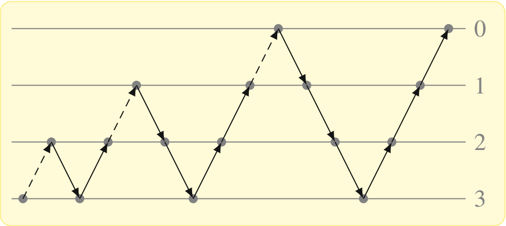


Fig. 4. Hybrid scheme with coarse-to-fine strategy

3 Hybrid Scheme for Energy Minimization

The energy function (4) now becomes

$$E = \int_{\Omega} \left[\varphi_d\left(\mathbf{x}, \mathbf{p}\right) + \alpha \varphi_s\left(\mathbf{x}, \mathbf{p}\right)\right] d\mathbf{x} \tag{9}$$

with the Euler-Lagrange equation system:

$$\begin{aligned} \varphi_d'\left(\bullet\right) f_d\left(\mathbf{x}\right) I_r I_x - \alpha \mathbf{div}\left(\varphi_s'(\bullet) f_s(\mathbf{x}) \nabla u\right) = 0 \\ \varphi_d'\left(\bullet\right) f_d\left(\mathbf{x}\right) I_r I_y - \alpha \mathbf{div}\left(\varphi_s'(\bullet) f_s(\mathbf{x}) \nabla v\right) = 0 \end{aligned} \tag{10}$$

where I_{*r} is the temporal difference

$$I_{*r} = I_{t+\Delta t,*}(\mathbf{x}+\mathbf{p}) - I_{t,*}(\mathbf{x}),$$

and I_* are the spatial derivatives in the next frame $I_{t+\Delta t,*}(\mathbf{x}+\mathbf{p})$. We choose the regularization functions φ_d, φ_s as $\varphi(s^2) = \sqrt{s^2+\varepsilon^2}$ that yields the total variation regularizer proposed in [13]. This regularizer leads to pseudo L_1-minimization. The quantity ε is chosen to be reasonably small, e.g. 0.001, to guarantee that φ is differentiable at $s = 0$.

The Euler-Lagrange equations are highly nonlinear due to the choice of φ_d and φ_s. The iteration scheme [12] is used to solve the flow field. It is necessary to approximate the global optimum of the energy using the iteration scheme and the multiscale

Algorithm 1. Mulrigrid scheme for flow estimation, V-cycle

```
if coarsest layer then
    Solve the flow field
else
    - Save result from previous step
    - Perform pre-relaxation on flow field
    - Restrict the flow to coarse layer
    Perform V-cycle on coarse layer
    - Calculate the error at coarse layer
    - Prolong the error to current layer
    - Update the flow at current layer
    - Perform post-relaxation on the flow
end if
```

approach. Let $\mathbf{p}^{(k)}$ be the flow field at step k, then the flow of the next iteration will be the solution of

$$\begin{aligned} 0 &= \varphi_d'(\bullet) f_d(\mathbf{x}) I_r^{(k+1)} I_x^{(k)} - \alpha \mathbf{div}\left(\varphi_s'(\bullet) f_s(\mathbf{x}) \nabla u^{(k+1)}\right) \\ 0 &= \varphi_d'(\bullet) f_d(\mathbf{x}) I_r^{(k+1)} I_y^{(k)} - \alpha \mathbf{div}\left(\varphi_s'(\bullet) f_s(\mathbf{x}) \nabla v^{(k+1)}\right) \end{aligned} \tag{11}$$

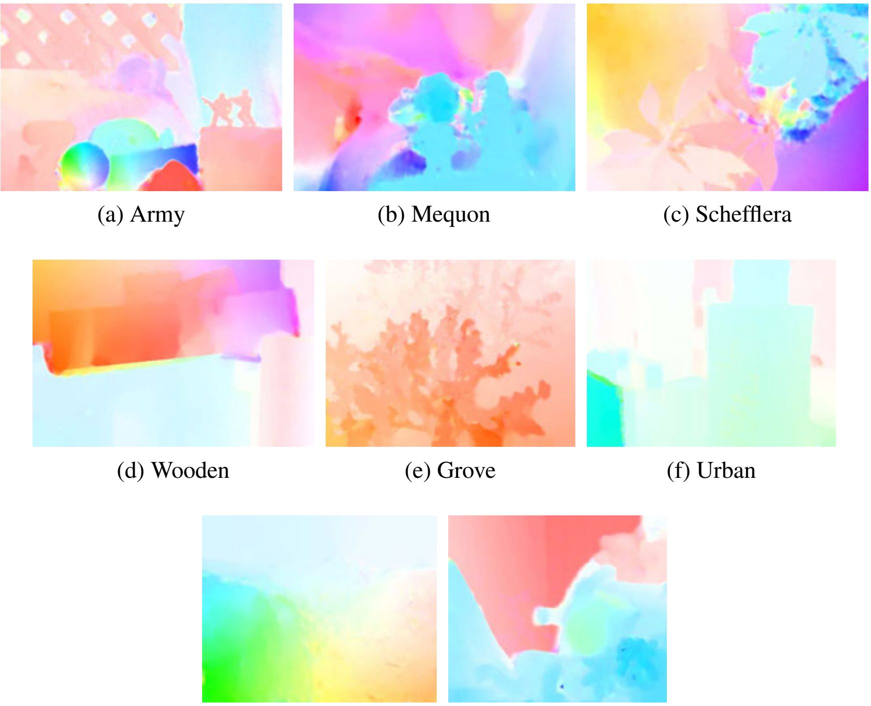
where

$$\begin{aligned} \varphi_d'(\bullet) &= \varphi'\left(f_d(\mathbf{x})\left(|I_{t+\Delta t}(\mathbf{x}+\mathbf{p})^{(k+1)} - I_t(\mathbf{x})|^2\right)\right), \\ \varphi_s'(\bullet) &= \varphi'\left(f_s(\mathbf{x})\left(|\nabla u^{(k+1)}|^2 + |\nabla v^{(k+1)}|^2\right)\right). \end{aligned}$$

The details of the discretization form can be derived easily, so we do not show them here. Both the coarse-to-fine scheme [3,5,12,14] and the multigrid scheme [11,15,16,17] have been used so far to solve (11) effectively. Here, we introduce the hybrid scheme to take advantage of the multigrid scheme and the coarse-to-fine scheme to produce an effective solver. The hybrid scheme is designed to solve (11) with an arbitrary scale parameter.

The purpose of the proposed scheme is to cope with large motion and improve the robustness of the solver simultaneously. While large motion can be detected at a coarse scale, the sharpness and precision of the flow field are enhanced at a fine scale. We build the pyramid of images and its derivative with the scale parameter s that can be larger than 0.5. The larger the value of s, the higher the computation cost. The idea of the multigrid solver is to solve the residual equations at the coarse layer and prolong the error from the coarse layer to the fine layer to correct the flow field. The flow is incrementally updated each iteration step as we use the iteration scheme. Thus, we employ the idea of the multigrid solver and form the scheme in Fig. 3.

Let $u^i_{s^k h}$ be the flow value at iteration ith on the kth layer in the pyramid model. Fig. 3 shows the basic V-cycle that we use to solve equation system. First, we perform the



(a) Army (b) Mequon (c) Schefflera

(d) Wooden (e) Grove (f) Urban

(g) Yosemite (h) Teddy

Fig. 5. Visual result of our method on evaluation sequences from Middlebury's dataset [10]

relaxation step on flow $u^{i}_{s^k h}$ to yield $u^{i+1}_{s^k h}$. The flow $u^{i+1}_{s^k h}$ is restricted to a coarser layer, as $u^{i+1}_{s^{k+1} h}$ with the scale factor s. This pre-relaxation step and restriction step continue until we reach the coarsest layer. At this stage, we can simply calculate the error, as the difference between the final result and the restricted result from the fine layer. As we have the error at a certain layer, say $e^{i+2}_{s^{k+1} h}$, we can propagate it to a finer layer, as $e^{i+2}_{s^k h}$ with the scale factor s^{-1}. The flow is updated using this error as follows

$$u^{i+1}_{s^k h} \leftarrow u^{i+1}_{s^k h} + e^{i+2}_{s^k h}$$

We perform post-relaxation on the flow $u^{i+1}_{s^k h}$ once again, before calculating the error at the current layer and propagating it to the finer layer. This process repeats until we reach the finest layer. Fig. 3 and algorithm 1 summarize the details of this scheme.

The coarse-to-fine strategy is used along with the V-cycle that we described above to deal with large motion. We perform the V-cycle on each layer. The result is then propagated to the finer layer with scale factor s^{-1}. This process is repeated from the coarsest layer to the finest layer, as in Fig. 4. We can achieve a good result with only a few iterations at each relaxation step. In the experiment, we use five iterations for each relaxation step.

4 Experiments

The quality of flow field is evaluated by angular error and end-point error. Some other measurements are also used but they are not comparable in the context of object movement. The angular error is given as follows:

$$e_\theta = \arccos\left(\frac{\mathbf{p}^T\mathbf{p}_{gt}+1}{\sqrt{\mathbf{p}^T\mathbf{p}+1}\sqrt{\mathbf{p}_{pt}^T\mathbf{p}_{pt}+1}}\right) \tag{12}$$

where $\Delta t = 1$ and $\mathbf{p}_{gt} = (u_{gt}, v_{gt})$ is the true motion field of the current image.

Table 1. Estimation results on synthetic training sequences [10]

Sequence	AEE	STD	AAE	STD
Dimetrodon	0.154	0.154	2.667	2.598
Grove2	0.246	0.435	3.509	6.972
Grove3	0.687	1.424	6.910	16.613
Hydrangea	0.181	0.376	2.221	5.574
RubberWhale	0.128	0.324	4.247	11.799
Urban2	0.427	1.276	3.199	8.319
Venus	0.332	0.604	4.718	13.356

4.1 Synthetic Images

First, the experiments were performed on the training data with available groundtruths. Our proposed flow field is very sharp along the object boundary. Table 1 gives the quantitative evaluation, where AEE is the average end-point error, AAE is the average angular error, and STD is the standard deviation of those two errors. We perform experiments on these training sequences to get the parameter set that yield the most stable results through difference sequences. Even though the smooth parameter can be embedded inside the control function, we still keep it as additional parameter for our experiments.

The proposed model does not operate at its best, as we are using the grayscale image, since much information has been discarded. In addition, it is hard to specify which point belongs to object by its color, because the color range is limited on the grayscale image. Therefore, comparing our method to other methods operating on color images is unfair. However, even if the grayscale image limits our model, we still obtained some good results, as shown in Table 1.

We also performed the experiments with the evaluation dataset on Middlebury's website [10] to show how the proposed model can improve a given model. We start from a very basic model, which is close to the model in [14]. We even discard the gradient term in the model. We choose this basic model, as it can easily reveal the performance

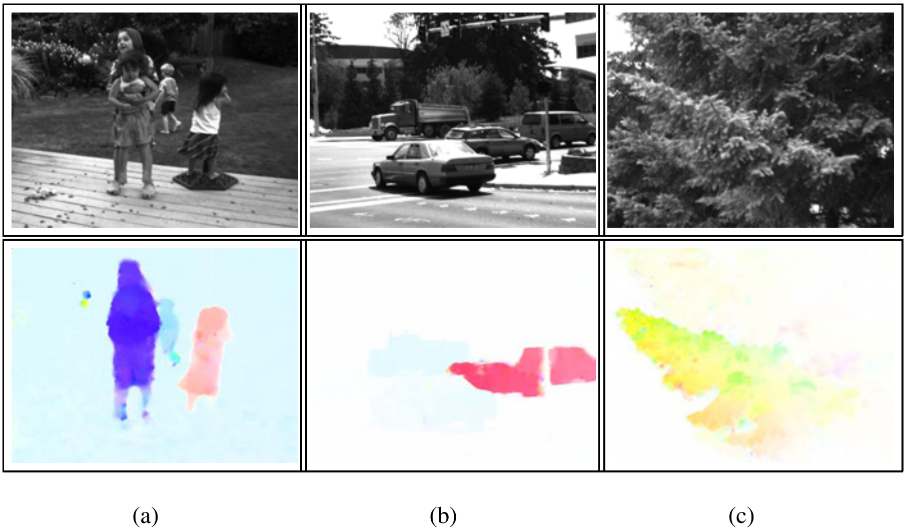


Fig. 6. Estimation result of our method on some evaluation sequences from MiddleBury's dataset [10]. First row is frame 10 and second row is the estimated flow field. (a) Backyard, (b) Dumptruck, (c) Evergreen.

of our proposed model and our optimization scheme. It is very hard to see how good the result is for advanced models in the top of Middlebury's list. The results at the top of the table are very close to one another.

Second column of Table 2 shows the results with the present of Hybrid scheme. The results in first column are obtained with the used of control functions and the multiscale approach. As we see here, the Hybrid scheme reduces the end-point error for most of evaluation sequences comparing to the traditional multiscale approach. The Hybrid scheme gives about the same performance as multiscale approach for Urban, Yosemite, and Teddy. However, it still improves the end-point error 9% accuracy in average (30% on Mequon sequence, 33% on Wooden sequence, 23% on Army). A similar observation can be seen in Table 3. It reduces 5% of angular error in average (31% on Mequon and Wooden sequence). As a result, the Hybrid scheme shows that it can cope with large motion as multiscale approach and slightly improves accuracy of the flow field. These results consistently show the improvement of our proposed method compared to the original model. Our method yields better results than some of the current state-of-art methods [4,5,18,19,20,21,22,23,24] for Urban sequence. A similar observation can be seen on the Grove sequence. The angular errors on Urban, Teddy, and Yosemite sequences are higher when we apply the Hybrid scheme. The reason for large error on these sequences comes from low contrast image (Urban, Yosemite) and low texture (top right side of Teddy). Thus, additional correction step can be applied to push the accuracy further.

Table 2. End-point error on evaluation sequences [10]

Sequence	with adapting functions	+ hybrid scheme
Army	0.22	0.17
Mequon	0.87	0.61
Schefflera	1.17	1.20
Wooden	0.99	0.66
Grove	1.17	0.99
Urban	0.72	0.73
Yosemite	0.14	0.18
Teddy	1.37	1.30

Table 3. Angular error on evaluation sequences [10]

Sequence	with adapting functions	+ hybrid scheme
Army	7.88	6.49
Mequon	13.4	9.22
Schefflera	17.6	16.4
Wooden	12.0	8.22
Grove	4.38	3.77
Urban	5.69	6.84
Yosemite	2.75	3.59
Teddy	6.59	7.51

4.2 Real-Life Images

We are interested in the results on real-life sequences for applications. Therefore, we performed the method on some real-life sequences to prove how effective it is for real applications. Middlebury's website has another measurement, termed interpolation error, beside the end-point error and angular error. This measurement, however, does not completely hold in our case. As we are interested in the object movement, rather than intensity matching, the interpolation error can be large due to the occlusion problem. Thus, the evaluation results on interpolation error cannot be compared in our case.

Fig. 6 shows some results on real-life sequences of the evaluation dataset [10]. The discontinuity of the flow field along the object boundary is highly preserved. Other experiments are performed on real-life sequences from the training dataset [10]. Fig. 7 shows some visual results on these sequences. The results on the DogDance sequence and MiniCooper sequence are proof for the effectiveness of our proposed method. The proposed method indeed improves the smoothness of flow field, while retaining the sharpness on edge in case of DogDance and MiniCooper sequences. This is especially helpful when we use the result for motion segmentation, object isolation, etc. We can see that our method outperforms the original model. We achieve the sharpness, the smoothness, and solve the occlusion problem simultaneously. Our proposed method shows how we can improve performance of a given estimation model further.

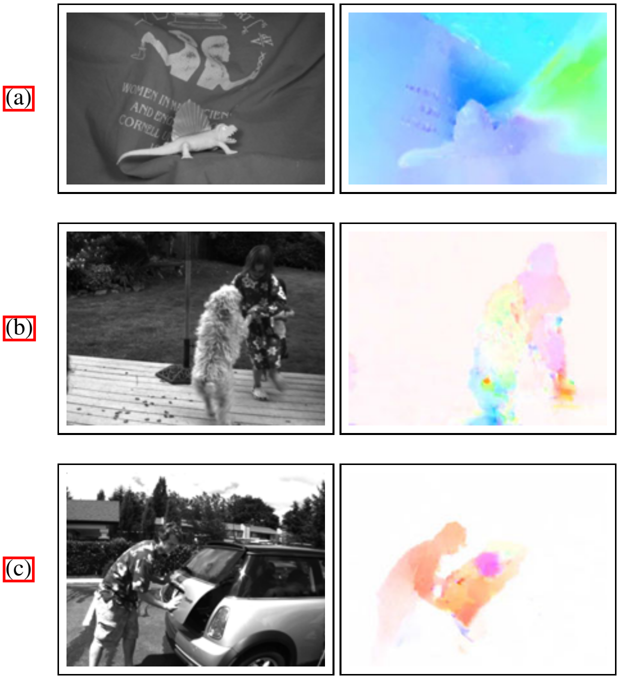


Fig. 7. Estimation result of our method on other real-life sequences [10]. (a) Dimetrodon, (b) DogDance, (c) MiniCooper.

5 Conclusion

We proposed an improved algorithm for optical flow estimation using the variational model. The image information was used to tune the estimation process. We introduced the adapting functions and embedded them to the energy function. Our model also addressed the occlusion problem. We proposed a hybrid scheme that took advantage of the coarse-to-fine approach and the multigrid solver to yield more robust results. The result with the present of hybrid solver and control functions was indeed much better and more robust.

We are applying the method for grayscale images. Further work includes the extension of our approach to color images. Our approach can also be integrated into advanced models to produce even better results. An in-depth study of the effect of our method on advanced models may lead to some interesting results in the near future.

References

1. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence 17(1-3), 185–203 (1981)
2. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI, pp. 674–679 (April 1981)

3. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
4. Xu, L., Chen, J., Jia, J.: A Segmentation Based Variational Model for Accurate Optical Flow Estimation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 671–684. Springer, Heidelberg (2008)
5. Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, A., Rosenhahn, B., Seidel, H.-P.: Complementary Optic Flow. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 207–220. Springer, Heidelberg (2009)
6. Wedel, A., Pock, T., Braun, J., Franke, U., Cremers, D.: Duality tv-l1 flow with fundamental matrix prior. In: Image and Vision Computing New Zealand (2008)
7. Werlberger, M., Pock, T., Bischof, H.: Motion estimation with non-local total variation regularization. In: CVPR, pp. 2464–2471 (2010)
8. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An Improved Algorithm for TV-L1 Optical Flow. In: Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R. (eds.) Statistical and Geometrical Approaches to Visual Motion Analysis. LNCS, vol. 5604, pp. 23–45. Springer, Heidelberg (2009)
9. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic huber-l1 optical flow. In: BMVC, British Machine Vision Association (2009)
10. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV, pp. 1–8 (2007)
11. Bruhn, A., Weickert, J.: Towards ultimate motion estimation: combining highest accuracy with real-time performance. In: ICCV, vol. 1, pp. 749–755 (October 2005)
12. Papenberg, N., Bruhn, A., Brox, T., Didas, S., Weickert, J.: Highly accurate optic flow computation with theoretically justified warping. Int. J. Comput. Vision 67(2), 141–158 (2006)
13. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D 60(1-4), 259–268 (1992)
14. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. Int. J. Comput. Vision 61(3), 211–231 (2005)
15. Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., Schnörr, C.: Variational optic flow computation in real-time. IEEE Trans. Image Proc. 14(5), 608–615 (2005)
16. Bruhn, A., Weickert, J., Kohlberger, T., Schnörr, C.: Discontinuity-Preserving Computation of Variational Optic Flow in Real-Time. In: Kimmel, R., Sochen, N.A., Weickert, J. (eds.) Scale-Space 2005. LNCS, vol. 3459, pp. 279–290. Springer, Heidelberg (2005)
17. Bruhn, A., Weickert, J., Kohlberger, T., Schnörr, C.: A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. Int. J. Comput. Vision 70(3), 257–277 (2006)
18. Lei, C., Yang, Y.-H.: Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In: ICCV, pp. 1562–1569 (2009)
19. Lee, K.J., Kwon, D., Yun, I.D., Lee, S.U.: Optical flow estimation with adaptive convolution kernel prior on discrete framework. In: CVPR, pp. 2504–2511. IEEE (2010)
20. Lempitsky, V.S., Roth, S., Rother, C.: Fusionflow: Discrete-continuous optimization for optical flow estimation. In: CVPR (2008)
21. Glocker, B., Paragios, N., Komodakis, N., Tziritas, G., Navab, N.: Optical flow estimation with uncertainties through dynamic mrfs. In: CVPR (2008)
22. Seitz, S.M., Baker, S.: Filter flow. In: ICCV, pp. 143–150 (2009)
23. Sun, D., Roth, S., Lewis, J.P., Black, M.J.: Learning Optical Flow. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 83–97. Springer, Heidelberg (2008)
24. Li, Y., Huttenlocher, D.P.: Learning for Optical Flow Using Stochastic Optimization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 379–391. Springer, Heidelberg (2008)

Real-Time Background Compensation for PTZ Cameras Using GPU Accelerated and Range-Limited Genetic Algorithm Search

Thuy Tuong Nguyen and Jae Wook Jeon*

School of Information and Communication Engineering,
Sungkyunkwan University, Suwon, Korea
ntthuy@skku.edu, jwjeon@yurim.skku.ac.kr

Abstract. We propose a range-limited Genetic Algorithm (GA) search with an accelerated Graphics Processing Unit (GPU) based implementation for background compensation where pan-tilt-zoom (PTZ) cameras are used. Our method contains GA with search ranges restricted using histogram matching and GPU implementation of the range-limited GA. First, based on histogram matching, estimation of approximate scale (camera zoom) and translation (camera pan and tilt) parameters is used to restrict the ranges for the later GA search. Next, the GA is applied to find an optimal solution. Experimental comparisons of the proposed method to existing methods show that our work has advantages: robust to critical situations due to using GA, and fast processing.

Keywords: Background compensation, histogram matching, GA, GPU.

1 Introduction

Intelligent visual surveillance, using computer vision techniques, is increasingly important over recent years. PTZ cameras are mainly used in this research area for object detection and tracking. The challenge of using this type of camera is to eliminate the dynamic background caused by camera motion. Hence, motion based tracking with a PTZ camera encounters difficulties such as identifying features of object motion, compensating background motion, and tracking mechanism. Although compensating background requires less computational cost and memory storage compared to mosaicing background and optical flow clustering [1], it typically yields a non-optimal solution.

Alternative approaches have been proposed to compensate background. In [2], specialized hardware is used to measure pan, tilt and zoom parameters. Relationship between pixels representing the same 3-D point in frames is estimated

* This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-(C1090-1121-0008)), and by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0018397).

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 85–96, 2011.

to eliminate background motion. Background motion is represented by an affine transformation in [3], and affine motion parameters are estimated using least median of squares. In this method, a number of feature points in the current image and their corresponding points in the previous image are picked out, and the pixels from the regions of moving objects are considered as outliers of the affine estimator. However, it is sensitive in selecting feature points due to motion blur. In [4], M-estimator like techniques in a multiresolution framework are used as the parametric motion model estimation. Multi-resolution Hough transform is used to estimate affine parameters in [5]. Results in this paper show that the main advantage resides with motion estimation in presence of motion blur. Overcoming disadvantanges of the previous methods, [1] presents the 1-D feature matching and outlier rejection method. This method is robust to motion blur and moving object proportion.

GA have been widely applied in estimating global motion. It is a stochastic search technique based on the principles of natural selection and genetics to find the approximate solutions of optimization and search problems. A GA in the continuous space to estimate global motion is proposed in [6]. A multiresolution GA is proposed in [7] to solve the imagerelated optimization problems image segmentation, stereo vision and motion estimation. Background motion can be effectively determined based on the proposed motion estimation method. For a problem, GA can yield a near-optimal solution when it is well-modeled and its related parameters are appropriately configured. However, there is a trade-off between computational cost and accuracy: higher numbers of chromosomes and generations increase processing time. Therefore, the existing GA based background compensation methods are not appropriate for real-time systems when high accuracy is being considered.

Range-limited GA search implemented using GPU is presented in this paper to achieve high accuracy and low processing time of background compensation for PTZ cameras. Our method overcomes all drawbacks of existing methods: motion blur, a large proportion of moving objects, and difficulty in selecting feature points. The proposed GA model contains two components: 1) estimation of approximate scale (camera zoom) and translation (camera pan and tilt) parameters is used to limit ranges for the later GA search, and 2) GA search is used to reach the optimal solution. The estimation model employs projection histograms to quickly determine the scale and translation parameters. Furthermore, processing time is significantly reduced by the GPU implementation.

Our method of range-limited GA search for background compensation is presented in Section 2. GPU implementation is in Section 3. Section 4 presents experimental results. The paper is drawn to a conclusion in Section 5.

2 Range-Limited GA Search

2.1 Background Motion Model and GA Search

GA typically contains a population of suitably encoded solutions to the problem and contains an evaluation function. It differs from other traditional optimization techniques, because it involves a search from a population of solutions, but not from a single point. Due to its characteristics, GA is perfectly suited to solving the optimization problem represented by background compensation.

Based on [1], we initially assume that the relationship between successive frames captured by a PTZ camera can be approximated by a transformation of four parameters (two directional zooms, pan and tilt). Hence, a chromosome in the GA based background compensation problem can be identified with the array formed by motion model parameters. In the case of an affine motion model,

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{b}, \tag{1}$$

the chromosome is thus formed by the combination of four, in this paper, motion parameters. This means $\mathbf{A}$ and $\mathbf{b}$ are 2×2 and 2×1 matrices, respectively:

$$\mathbf{A} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}; \mathbf{b} = \begin{pmatrix} t_x \\ t_y \end{pmatrix}; s_x, s_y > 0. \tag{2}$$

The chromosome representing four parameters, s_x, s_y, t_x and t_y, are floating point vectors. In GA, a random initial population with a given number of individuals is first generated. If the explicit information of the system is provided in advance, then such knowledge can be included in the initial population (e.g., we propose a method to restrict the search range of scale and translational parameters). In the second step of GA, the fitness of each individual is evaluated. The fitness function is defined for background compensation as:

$$f\left(C_j^i\right) = \frac{1}{\eta} \sum_{\mathbf{x}} |I_t(\mathbf{x}) - Aff\left(I_{t-1}(\mathbf{x})\right)|, \tag{3}$$

where $C_j^i = \left((s_x)_j^i, (s_y)_j^i, (t_x)_j^i, (t_y)_j^i\right)$ is the chromosome j ($j = 1, 2..., N_p$, where N_p is the number of chromosomes) at the ith generation; $I_t(\mathbf{x})$ and $I_{t-1}(\mathbf{x})$ are the pixel values of the point $\mathbf{x} = (x, y)$ in the frame t and $t-1$; Aff is the affine function utilizing $\left((s_x)_j^i, (s_y)_j^i, (t_x)_j^i, (t_y)_j^i\right)$; η is the number of common points in images I_t and I_{t-1}. The third step is selection; it is implicitly coupled with the replacement step. The fourth step is applying the crossover. Two chromosomes (parents) from the current population are randomly selected to be mated. In this paper, we apply uniform crossover to our problem, and we ignore the mutation step for simplicity but not for less effectiveness. All steps are repeated for each generation until a termination condition is met.

To speed up the GA process and to obtain high accuracy, we propose a method to limit the search ranges of scale and translation parameters in the motion model (1). Various studies, attempting to reduce computational cost, use techniques such as hibridizing [8], predicting [9] or imitating [10]. Users should have knowledge about the nonlinearities of real-world applications to define a trade-off between accuracy and computation speed. In this paper, we focus on a method that restricts GA search ranges.

2.2 Limiting the Search Range of Scale Parameters

This section presents the scale parameter estimation algorithm based on histogram matching. It is assumed that scale parameters, s_x and s_y, in the motion model (1) can be separately estimated from translation parameters, t_x and t_y. In this case, the translation matrix becomes $\hat{\mathbf{b}} = (0, 0)^T$, and the scale matrix $\hat{\mathbf{A}}$ is formed by $\hat{s}_x$ and $\hat{s}_y$. The estimated parameters, $\hat{s}_x$ and $\hat{s}_y$, are supposed to approximate the actual parameter, s_x and s_y, of the motion model. In fact, s_x and s_y might be inside the range $[\hat{s} - \Delta s; \hat{s} + \Delta s]$ where $\hat{s}$ represents $\hat{s}_x$ or $\hat{s}_y$, and Δs is a value used to define the search range for s_x and s_y around the anchor $\hat{s}$. Hence, the relationship between two images I_t and I_{t-1} is defined as:

$$I_t = I_{t-1}^{\hat{\mathbf{A}}} = \varsigma^{\hat{s}_x}\big(\varsigma^{\hat{s}_y}(I_{t-1})\big) = \varsigma^{\hat{s}_y}\big(\varsigma^{\hat{s}_x}(I_{t-1})\big), \tag{4}$$

where the scale parameter $\hat{\mathbf{A}}$ is decomposed into two directional scale values, one represents the scale with respect to the horizontal direction, the other is for the vertical direction; $\varsigma^{\hat{s}_x}(I_{t-1})$ and $\varsigma^{\hat{s}_y}(I_{t-1})$ are the horizontal and vertical scaling functions of I_{t-1}. The scaling-down functions resize the full-size image I_{t-1} to an image region in I_t, i.e., the scaled I_{t-1} stays inside I_t. Similarly, the scale-up functions resize the full-size image I_t to an image region in I_{t-1}.

We simplify estimating two scale parameters to estimating only $\hat{s}_y$, then we seek for the optimal s_x and s_y inside the range $[\hat{s}_y - \Delta s; \hat{s}_y + \Delta s]$. There are two reasons: 1) it is to reduce the complexity of estimation with an assumption that camera zoom is homogeneous in both horizontal and vertical directions; 2) estimating $\hat{s}_y$ but not $\hat{s}_x$ is to achieve high performance memory access in GPU implementation with the usage of coalesces.

In this paper, vertical and horizontal histograms of gray images are constructed. The matching value of two histograms is calculated based on values of the two histograms. The vertical histogram of an image is constructed by projecting along vertical lines (columns), and one histogram is constructed per vertical line (column). Let $H^V(I)$ be the vertical histogram of the image I:

$$H^V(I) = \big\{h_j^V(I) : j = 0, 1, ..., W - 1\big\}, \tag{5}$$

where W is the number of columns; $h_j^V(I)$ is the histogram of column j of I:

$$h_j^V(I) = \big\{h_{jl}^V(I) : l = 0, 1, ..., L - 1\big\}, \tag{6}$$

where L is the number of bins; $h_{jl}^V(I)$ is the number of pixels with intensities in bin l. The matching value $d^V(.)$ of vertical histograms $h_i^V(I_{t-1}^{\mathbf{r}})$ and $h_j^V(I_t^{\mathbf{r}})$ is

$$d^V\big(h_i^V(I_{t-1}^{\mathbf{r}}), h_j^V(I_t^{\mathbf{r}})\big) = 1 - \xi_4/(2H), \tag{7}$$

where $\xi_4 = \sum_{l=0}^{L-1} \big|h_{il}^V(I_{t-1}^{\mathbf{r}}) - h_{jl}^V(I_t^{\mathbf{r}})\big|$, $I_{t-1}^{\mathbf{r}} = I_{t-1}$, $I_t^{\mathbf{r}}$ is an image region inside I_t if camera zooms out. Similarly, $I_t^{\mathbf{r}} = I_t$, and $I_{t-1}^{\mathbf{r}}$ is an image region of I_{t-1}

when camera zooms in. Let $D^V\left(H^V(I_{t-1}), H^V(I_t), \hat{s}_y\right)$ be the matching value of two vertical histograms of I_{t-1} and I_t:

$$D^V\left(H^V(I_{t-1}), H^V(I_t), \hat{s}_y\right) = \xi_5/(W-1), \tag{8}$$

where ξ_5 is calculated by: If $\hat{s}_y < 1$ (zoom out), $\xi_5 = \sum_{i=0}^{W-1} d^V\left(h_i^V(I_{t-1}), h_i^V(I_t^{\mathbf{r}})\right)$, and if $\hat{s}_y \geq 1$ (zoom in), $\xi_5 = \sum_{i=0}^{W-1} d^V\left(h_i^V(I_{t-1}^{\mathbf{r}}), h_i^V(I_t)\right)$, where $I_{t-1}^{\mathbf{r}}$ and $I_t^{\mathbf{r}}$ are image regions of I_{t-1} and I_t. Scale ratio between two images with respect to vertical direction is determined by searching for the value $\hat{s}_y$ that maximizes the matching value of the vertical projection histograms of two images. Thus, $\hat{s}_y$ is

$$\hat{s}_y = \arg \max_{s_{min} < \hat{s}'_y < s_{max}} D^V\left(H^V(I_{t-1}), H^V(I_t), \hat{s}'_y\right), \tag{9}$$

Fig. 1 shows vertical scaling of the ith column in I_{t-1} and I_t. In case of camera zoom-out, I_t is approximately I_{t-1} added with two marginal regions. Similarly, the case of zoom in, I_{t-1} is approximately I_t plus two marginal sub-images.

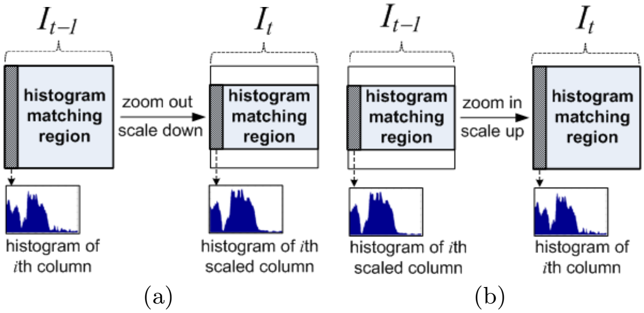

I_{t-1}
I_t
I_{t-1}
I_t
histogram matching region
zoom out
scale down
histogram matching region
histogram matching region
zoom in
scale up
histogram matching region
histogram of ith column
histogram of ith scaled column
histogram of ith scaled column
histogram of ith column
(a)
(b)

Fig. 1. Vertical scaling of a column in two successive images

2.3 Limiting the Search Range of Translation Parameters

This section presents the translation parameters estimation based on histogram matching. It is assumed that the translation parameters, t_x and t_y, in the motion model (1) can be separately estimated from the scale parameters, s_x and s_y. The goal is to estimate $\hat{\mathbf{b}}$ and then to define the GA search range of $\mathbf{b}$ before executing GA for background compensation. The estimated parameter, $\hat{\mathbf{b}}$, is supposed to approximate the actual parameter, $\mathbf{b}$, of the affine model. Indeed, $\mathbf{b}$ might be inside the range $[\hat{\mathbf{b}} - \mathbf{\Delta b}; \hat{\mathbf{b}} + \mathbf{\Delta b}]$ where $\varDelta\mathbf{b} = [\varDelta t_x, \varDelta t_y]^T$ is a value used to define the search range for $\mathbf{b}$ around the anchor $\hat{\mathbf{b}}$.

Similar to scale parameter estimation, the translational displacement between I_{t-1} and I_t in the x-axis direction is determined by searching for the value $\hat{t}_x$ that maximizes the matching value of the vertical histograms of the two images:

$$\hat{t}_x = \arg \max_{-W < \hat{t}'_x < W} D^V\left(H^V\left(I_{t-1}\right), H^V\left(I_t\right), \hat{t}'_x\right). \tag{10}$$

Displacement in y-axis direction is determined by searching for the value $\hat{t}_y$ that maximizes the matching value of the horizontal histograms of the two images:

$$\hat{t}_y = \arg \max_{-H<\hat{t}'_y<H} D^H \left(H^H \left(I_{t-1} \right), H^H \left(I_t \right), \hat{t}'_y \right). \quad (11)$$

where $H^H(I)$ is the horizontal histogram of the image I; $D^H(\cdot)$ is the matching value of two horizontal histograms.

Fig. 2(a) shows horizontal translation of the ith column in I_{t-1} to the $(i+\hat{t}'_x)$th column in I_t with a horizontal displacement $\hat{t}'_x$. Histograms of the two corresponding columns are matched with respect to the common histogram matching region (bold border rectangles) of the two images. Similarly, Fig. 2(b) shows the jth row in I_{t-1} is vertically translated by a displacement $\hat{t}'_y$ to the $(j+\hat{t}'_y)$th row in I_t. With every trial displacement $\hat{t}'_x$ (or $\hat{t}'_y$), the matching value of two vertical (or horizontal) histograms is calculated; later, the best match is found using (10) (or (11)).

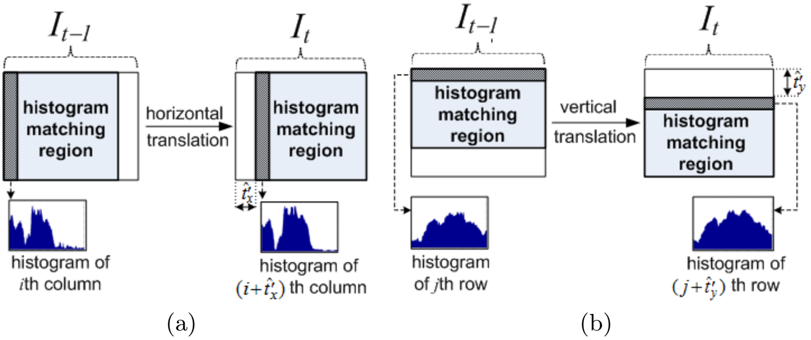

I_{t-1}
I_t
histogram matching region
horizontal translation
histogram matching region
$\hat{t}'_x$
histogram of ith column
histogram of $(i+\hat{t}'_x)$ th column
(a)
I_{t-1}
I_t
histogram matching region
vertical translation
histogram matching region
$\hat{t}'_y$
histogram of jth row
histogram of $(j+\hat{t}'_y)$ th row
(b)

Fig. 2. Horizontal and vertical translation of a column and a row in images

3 GPU Accelerated Implementation

In recent years, processing ability of GPU has rapidly increased. NVIDIA has developed the CUDA (compute unified device architecture) technology to indicate the problems of GPU. In this paper, the proposed range-limited GA search is implemented using CUDA. Fig. 3 summaries the high-level implementation architecture with <<< *kernel* >>> represents a function that is callable from the host (CPU) and executed on the GPU device simultaneously by parallel threads. Stages of GPU implementation are described as below.

3.1 Implementation of Translation Parameter Estimation

Initializing and Calculating $H^V(I_{t-1})$, $H^V(I_t)$, $H^H(I_{t-1})$ and $H^H(I_t)$. $H^V(I_{t-1})$, $H^V(I_t)$, $H^H(I_{t-1})$ and $H^H(I_t)$ are all set to zeros. In our implementation, N_{bins}, the number of bins in a histogram, is equal to the number of threads;

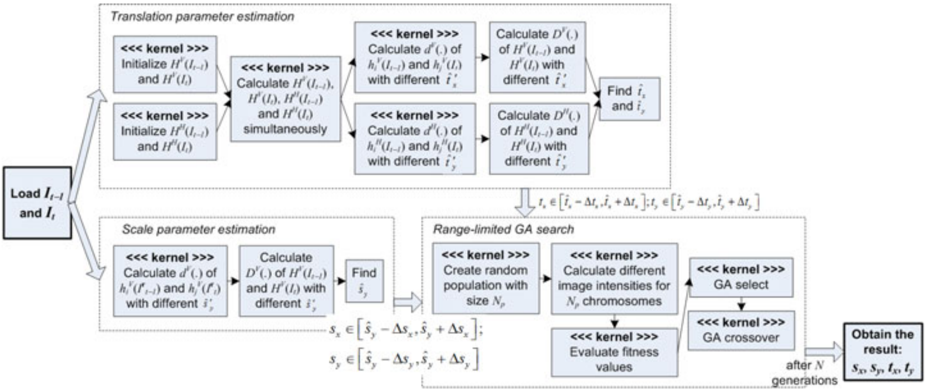

Translation parameter estimation
<<< kernel >>> Initialize $H^V(I_{t-1})$ and $H^V(I_t)$
<<< kernel >>> Initialize $H^H(I_{t-1})$ and $H^H(I_t)$
<<< kernel >>> Calculate $H^V(I_{t-1})$, $H^V(I_t)$, $H^H(I_{t-1})$ and $H^H(I_t)$ simultaneously
<<< kernel >>> Calculate $d^V(.)$ of $h_i^V(I_{t-1})$ and $h_j^V(I_t)$ with different $\hat{t}'_x$
<<< kernel >>> Calculate $d^H(.)$ of $h_i^H(I_{t-1})$ and $h_j^H(I_t)$ with different $\hat{t}'_y$
Calculate $D^V(.)$ of $H^V(I_{t-1})$ and $H^V(I_t)$ with different $\hat{t}'_x$
Calculate $D^H(.)$ of $H^H(I_{t-1})$ and $H^H(I_t)$ with different $\hat{t}'_y$
Find $\hat{t}_x$ and $\hat{t}_y$
Load I_{t-1} and I_t
$t_x \in [\hat{t}_x - \Delta t_x, \hat{t}_x + \Delta t_x]; t_y \in [\hat{t}_y - \Delta t_y, \hat{t}_y + \Delta t_y]$
Scale parameter estimation
<<< kernel >>> Calculate $d^V(.)$ of $h_i^V(I'_{t-1})$ and $h_j^V(I'_t)$ with different $\hat{s}'_y$
Calculate $D^V(.)$ of $H^V(I_{t-1})$ and $H^V(I_t)$ with different $\hat{s}'_y$
Find $\hat{s}_y$
$s_x \in [\hat{s}_y - \Delta s_x, \hat{s}_y + \Delta s_x]$;
$s_y \in [\hat{s}_y - \Delta s_y, \hat{s}_y + \Delta s_y]$
Range-limited GA search
<<< kernel >>> Create random population with size N_p
<<< kernel >>> Calculate different image intensities for N_p chromosomes
<<< kernel >>> Evaluate fitness values
<<< kernel >>> GA select
<<< kernel >>> GA crossover
after N generations
Obtain the result: s_x, s_y, t_x, t_y

Fig. 3. High-level GPU implementation architecture of the proposed range-limited GA

and the number of thread blocks corresponds to $imgHeight$ (or $imgWidth$ in the case of horizontal histograms). Next, pixels from two images are accumulated to vertical and horizontal histograms. At each pixel position, the gray value is first stored to a vertical histogram, then it is accumulated to a horizontal histogram. Hence, a thread is responsible for accumulating two pixel values from two images to projection histograms. The projection histograms, stored in global memories, are distributed among threads. Updating those histograms is data dependent, since many threads might attempt to update the same memory location. This situation results in writing conflicts. The conflict is effective solved with atomic operators [11] used in device functions.

Calculating Matching Values with Different Displacements. The kernel requires a minimum number of threads, $N_t = 16$, and a large number of blocks (e.g., in vertical histogram matching, the number of blocks is $(imgWidth \times numberOfDisplacements)/N_t$). The global index belonging to this kernel, calculated using built-in variables $blockIdx$, $blockDim$ and $threadIdx$, is used to encode the succession of displacements $\hat{t}'_x$ (or $\hat{t}'_y$) and ith column (or row). Hence, differently matching values of every pair of vertical (or horizontal) histograms are calculated and stored in an array of floating point numbers. We notice that sequentially changing the displacements $\hat{t}'_x$, $\hat{t}'_y$ are replaced by simultaneously processing a number of displacements in only one kernel. Thus, there are no loops in kernel code. Afterwards, the best matches $\hat{t}_x$ and $\hat{t}_y$ are found.

3.2 Implementation of Scale Parameter Estimation

This section presents the vertically scaled histogram matching. Similar to the previous section, however, columns in images are scaled and matched in this case. For simplicity, the nearest-neighbor interpolation method is applied to resize image columns. Distinct from separating codes into three kernels in the previous section, the code in this section is integrated into one kernel with the optimized utilization of shared memories. Steps of initializing, accumulating histograms (with interpolation), and calculating matching values are completely manipulated on shared memories. Reasons of this implementation are: 1) only vertical histograms are considered (each image pixel is essentially visited once), 2) based on the first reason, memory conflicts are eliminated in this implementation: one thread processes one column and calculates the corresponding matching value of two vertically scaled histograms, and 3) shared memories, which are potentially 150× faster than global memories and can be as fast as registers, are utilized.

3.3 Implementation of GA

Random Population Generation. A random initial population is generated with a given number of chromosomes. A efficient parallel random number generator [12] is utilized for our purpose. Each initial chromosome is formed by four floating point numbers that satisfy the conditions $s_x \in [\hat{s}_y - \Delta s_x, \hat{s}_y + \Delta s_x]$, $s_y \in [\hat{s}_y - \Delta s_y, \hat{s}_y + \Delta s_y]$, $t_x \in [\hat{t}_x - \Delta t_x, \hat{t}_x + \Delta t_x]$ and $t_y \in [\hat{t}_y - \Delta t_y, \hat{t}_y + \Delta t_y]$; therefore, there are $4N_p$ generated floating point numbers. Each thread is assigned to work on four numbers at once, and access the population as a vector of *float4* to minimize the number of memory accesses.

Fitness Evaluation. Fitness function is the average of the sum of absolute differences between two images. The kernel is launched with one thread per pair of pixels (one pixel is from I_{t-1}, one is from I_t). Each thread calculates the different intensity of $I_t(\mathbf{x})$ and $Aff(I_{t-1}(\mathbf{x})$ as in (3). With the specification of the GeForce GTX 460 used in this paper, our implementation can process the maximum number of pixels $N_t \times N_b = 1024 \times 65535$. For instance, with a fixed size of the two images 320×240, $(N_t \times N_b)/(320 \times 240) \approx 873$ chromosomes can be computed in parallel. We work on $float4$ and compute the final value of each individual's fitness via parallel reduction [13].

Selection and Crossover. The grid of selection and crossover kernels depends on population size. In selection, we define the number of threads equal to population size, a multiple of 16 (to optimize, especially for SIMD-type processing). This means only one block is required. Similarly, in crossover, one block is used.

4 Experimental Results

Eight image sequences were used in experiments. These sequences were acquired by a PTZ camera while tracking moving objects. The camera used in the experiment was a Sony EVI-D100 CCD video camera. The captured images had a resolution of 320 × 240 pixels. Table 1 specifies their corresponding descriptions, such as environment, lighting condition, background complexity, and distance from camera to object. Fig. 4 shows sample images of test sequences.

We used a PC with an AMD Athlon ×2 Processor 2.90 GHz and 4–GB RAM, and NVIDIA GeForce GTX 460. We compared the error (intensity difference

Table 1. Description of test image sequences

	Indoor/Outdoor	Lighting condition	Background complexity	Distance to object	# images
SQ1	Indoor	Normal	High	4-5m	150
SQ2	Outdoor	Bright	High	10-15m	190
SQ3	Indoor	Dark	Low	11-15m	250
SQ4	Indoor	Dark	Medium	0-7m	320
SQ5	Indoor	Dark	Low	8-32m	340
SQ6	Outdoor	Normal	High	5-7m	440
SQ7	Indoor	Dark	High	4-16m	480
SQ8	Outdoor	Bright	High	50-100m	700

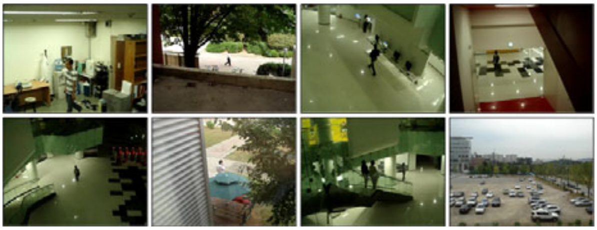


Fig. 4. Sample images of test sequences and correspondences. First row: SQ1, SQ2, SQ3, SQ4. Second row: SQ5, SQ6, SQ7, SQ8.

mean) and execution time of our method with other four methods: two Motion2D methods (M2D_ARN and M2D_AH2N) [4], multi-resolution Hough transform (MHT) [5], and Suhr's method [1]. For MHT, we set parameters as same as in [5]: number of binary images was 32, tolerance value was 0.1, scale range was $[0.9, 1.1]$, $\Delta\theta = \Delta\rho = 0.0003$, and reduction factor $\sigma = 2$, $\mu = 1$. With the method in [1], we set the same parameter values: $t_x, t_y \in [-25, 25]$ with 1 pixel resolution, $s \in [0.9, 1.1]$ with 0.01 resolution, 7 and 5 sub-images for horizontal and vertical feature extraction, window size for local minima and maxima was 3, matching threshold was 10, and matching search radius was 40 pixels.

We set the parameters so they were as similar to those of the other methods as possible. We defined $\hat{t}_x, \hat{t}_y \in [-25, 25]$ with 1 pixel resolution, and $\hat{s}_y \in [0.9, 1.1]$ with 0.01 resolution. GA parameters were: $N_p = 32$, crossover rate $p_c = 0.5$, and number of generations $n_{gen} = 5$. GA search ranges were defined as $\Delta t_x = \Delta t_y = 2$ and $\Delta s_x = \Delta s_y = 0.02$. It was noticed that $\hat{t}_x$, $\hat{t}_y$ and $\hat{s}_y$ were determined using histogram matching; therefore, $s_x \in [\hat{s}_y - \Delta s_x, \hat{s}_y + \Delta s_x]$, $s_y \in [\hat{s}_y - \Delta s_y, \hat{s}_y + \Delta s_y]$, $t_x \in [\hat{t}_x - \Delta t_x, \hat{t}_x + \Delta t_x]$ and $t_y \in [\hat{t}_y - \Delta t_y, \hat{t}_y + \Delta t_y]$. Table 2 shows the experimental results. Our method outperformed the other methods, especially in the critical cases of small difference in background and motion blur. Fig. 5 shows three typical cases of compensation results with comparing the proposed method to the others. Moreover, our processing time was stable to different image sequences where panning, tilting, and zooming have large changes.

Robustness of our method was evaluated in two ways. First, the error was analyzed with respect to the proportion of background with small difference. Based on Table 1, SQ5 and SQ8 were selected in the criterion of low background

Table 2. Error and time (ms) comparison of our method and others

	M2D_ARN		M2D_AH2N		MHT		Suhr's		Ours	
	Error	Time	Error	Time	Error	Time	Error	Time	Error	Time
SQ1	2.77	23.2	2.76	25.9	3.7	999.6	2.94	25.6	**2.61**	**15.6**
SQ2	2.43	30.6	2.42	34.9	3.18	843.9	3.37	33.7	**2.33**	**15.6**
SQ3	1.68	37.7	1.68	41.8	2.04	797.3	2.2	31.0	**1.58**	**16.0**
SQ4	1.5	51.0	1.49	53.6	1.9	649.8	1.69	27.2	**1.4**	**15.7**
SQ5	1.41	53.3	1.4	58.4	1.69	619.1	2.29	33.0	**1.34**	**15.4**
SQ6	3.41	71.7	3.4	77.5	4.52	903.2	5.17	28.4	**3.27**	**15.4**
SQ7	1.72	71.2	1.72	76.8	1.99	867.2	1.91	28.2	**1.63**	**15.4**
SQ8	2.24	108.8	2.23	122.2	2.83	956.6	2.26	30.4	**2.09**	**15.3**
Avg.	2.0	68.2	2.0	73.1	2.49	798.7	2.49	30.6	**1.9**	**15.6**

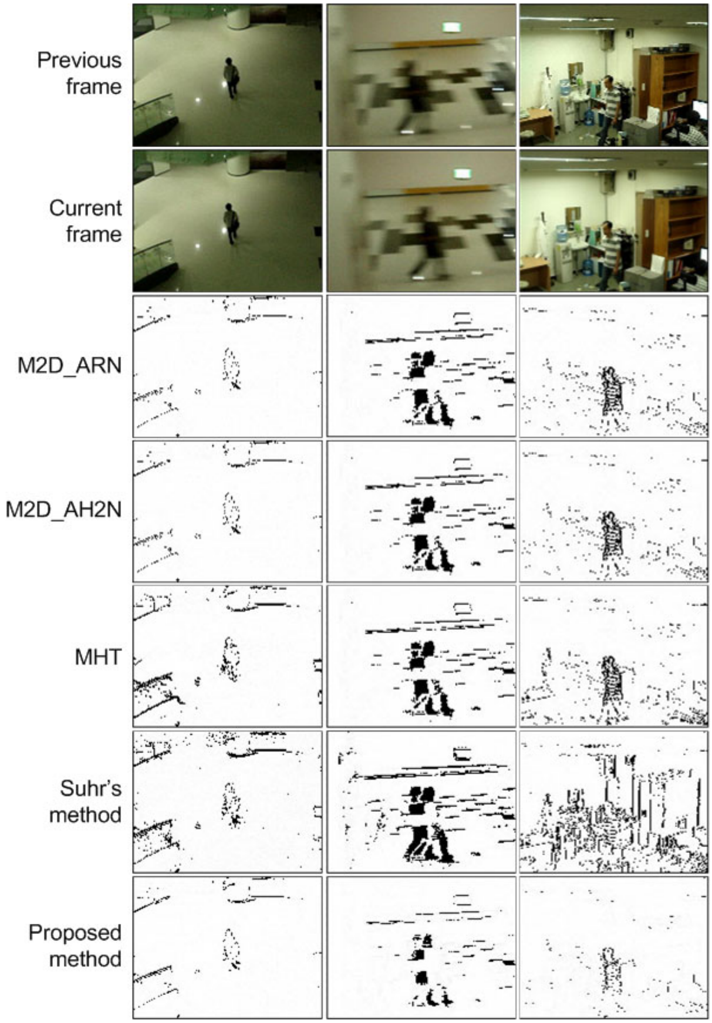
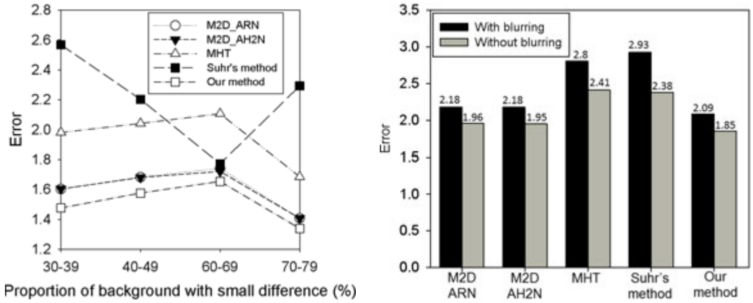


Fig. 5. Results of three typical examples of background compensation. From left to right: small different in background, severe blur, and zoom-in.

complexity. These sequences were corresponding to the background proportion 30–39%, 60–69%, 40–49%, and 70–79%. Fig. 6(a) shows error with different proportion of background with small difference. Suhr's method was not stable because the Hough space used to reject outliers had a large bin size. Second, the error was analyzed in terms of motion blurring effect. There were 530 blurred images manually selected from all sequences for this experiment. Fig. 6(b) presents the error comparison to blurring effect.



(a) (b)

Fig. 6. (a) Error with different proportion of background with small difference; (b) Error comparison to blurring effect

Table 3 presents time percentages of GPU processing. Code to estimate scale parameter were merged into one kernel to achieve full optimization; therefore, its processing time is lowest. Kernels of GA search occupied nearly all GPU time. The fitness evaluation kernel took most of GA time.

Table 3. GPU time percentages of our implemented kernels

Translation parameter estimation	4.516 %	Initialize $H^V(.)$ and $H^H(.)$	0.714 %
		Calculate $H^V(.)$ and $H^H(.)$	14.288 %
		Calculate $d^V(.)$ and $d^H(.)$	84.998 %
Scale parameter estimation	3.226 %	Initialize and calculate scaled histogram matching in one kernel	100 %
Range limited GA search	92.258 %	Create population	3.497 %
		Evaluate fitness	92.702 %
		Select	2.524 %
		Crossover	1.277 %

5 Conclusion

Our GPU accelerated and range-limited GA search is faster than the Motion2D methods by 4.5 times, the MHT method by 53 times, and the Suhr's method by 2 times. The computational improvement of our method is not only due to limiting the GA search ranges but also due to the GPU implementation techniques.

In this paper, the combination of GA and GPU implementation techniques is successfully applied to the background compensation problem for PTZ cameras. By experimentally comparing the results of our method to other methods, our work has two advantages. First, our method is robust in coping with critical situations, because the GA was proved that it can reach an optimal solution. Second, its processing time is very fast with the graphics card based implementation.

We plan to port the current implementation to a NVIDIA Tegra SoC mobile processor [14] to achieve high applicability on mobile computing platforms.

References

1. Suhr, J.K., Jung, H.G., Li, G., Noh, S.I., Kim, J.H.: Background Compensation for Pan-Tilt-Zoom Cameras using 1-D Feature Matching and Outlier Rejection. IEEE Trans. Circuits Syst. Video Technol. 21(3), 371–377 (2011)
2. Murray, D., Basu, A.: Motion Tracking with an Active Camera. IEEE Trans. Pattern Anal. Mach. Intell. 16(5), 449–459 (1994)
3. Araki, S., Matsuoka, T., Yokoya, N., Takemura, H.: Real-time Tracking of Multiple Moving Object Contours in a Moving Camera Image Sequence. IEICE Trans. Inform. Syst. E83–D(7), 1583–1591 (2000)
4. Odobez, J., Bouthemy, P., Temis, P.: Robust Multi-resolution Estimation of Parametric Motion Models in Complex Image Sequences. J. Vis. Commun. Image Represent. 6, 348–365 (1995)
5. Pham, X.D., Cho, J.U., Jeon, J.W.: Background Compensation using Hough Transformation. In: IEEE Int. Conf. Robot. Autom., pp. 2392–2397 (2008)
6. Moscheni, F., Vesin, J.: A Genetic Algorithm for Motion Estimation. In: 15eme Colloque sur le Traitement des Signaux et Images, France, pp. 825–828 (1995)
7. Gong, M., Yang, Y.H.: Quadtree-Based Genetic Algorithm and Its Applications to Computer Vision. Pattern Recognit. 37(8), 1723–1733 (2004)
8. Rodehorst, V., Hellwich, O.: Genetic Algorithm Sample Consensus (GASAC) - A Parallel Strategy for Robust Parameter Estimation. In: IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshop, pp. 103–110 (2006)
9. Mutoh, A., Nakamura, T., Kato, S., Itoh, H.: Reducing Execution Time on Genetic Algorithm in Real-World Applications using Fitness Prediction: Parameter Optimization of SRM Control. In: IEEE Congress Evol. Comput., pp. 552–559 (2003)
10. Jin, Y., Sendhoff, B.: Reducing Fitness Evaluations using Clustering Techniques and Neural Network Ensembles. In: Genetic Evol. Comput. Conf., pp. 688–699 (2004)
11. Shams, R., Kennedy, R.A.: Efficient Histogram Algorithms for NVIDIA CUDA Compatible Devices. In: Int. Conf. Signal Process. and Commun. Syst., pp. 418–422 (2007)
12. Tzeng, S., Wei, L.Y.: Parallel White Noise Generation on a GPU via Cryptographic Hash. In: Symp. on Interactive 3D Graphics and Games, pp. 79–87 (2008)
13. Harris, M., Sengupta, S., Owens, J.D.: Parallel Prefix Sum (Scan) in CUDA. Chapter 39 – GPU Gems 3 (2008)
14. NVIDIA Tegra Developer Zone, http://tegradeveloper.nvidia.com/tegra

Audio-Visual Speech Recognition Based on AAM Parameter and Phoneme Analysis of Visual Feature

Yuto Komai, Yasuo Ariki, and Tetsuya Takiguchi

Graduate School of System Informatics, Kobe University
Rokkodaicho 1–1, Nada-ku, Kobe, Hyogo, 657–8501 Japan
komai@me.cs.scitec.kobe-u.ac.jp, {ariki,takigu}@kobe-u.ac.jp

Abstract. As one of the techniques for robust speech recognition under noisy environment, audio-visual speech recognition using lip dynamic visual information together with audio information is attracting attention and the research is advanced in recent years. Since visual information plays a great role in audio-visual speech recognition, what to select as the visual feature becomes a significant point. This paper proposes, for spoken word recognition, to utilize **c** combined parameter(combined parameter) as the visual feature extracted by Active Appearance Model applied to a face image including the lip area. Combined parameter contains information of the coordinate value and the intensity value as the visual feature. The recognition rate was improved by the proposed feature compared to the conventional features such as DCT and the principal component score. Finally, we integrated the phoneme score from audio information and the viseme score from visual information with high accuracy.

1 Introduction

Recently, various speech recognition technologies have been put to practical use by the development of speech recognition technologies. However, in current speech recognition technologies, there is a problem that the recognition performance remarkably decreases under noisy environment, and it becomes a significant problem in aiming at the practical use of speech recognition.

Then, as one of the techniques for robust speech recognition under noisy environment, audio-visual speech recognition using lip dynamic visual information together with audio information is attracting attention and the research is advanced in recent years.

In audio-visual speech recognition, there are mainly three integration methods; early integration[1] that connects the audio feature vector with the visual feature vector, late integration[2] that weights the likelihood of the result obtained by a separate process for audio and visual signals, and synthetic integration[3] that calculates product of output probability in each state and so on. The research to lip-reading only in the visual feature is actively advanced

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 97–108, 2011.

because the visual feature, of course the audio feature, greatly influences the recognition rate in these processing. As the visual feature, various techniques such as width and height of lip[4], optical flow[5] and DCT[6] are employed.

In our research, the lip area is automatically extracted by Active Appearance Models[7][8] (AAM) regardless of speaker's position in the dynamic scene. Moreover, the combined parameter of AAM(**c** parameter) is employed as the feature parameter for utterance recognition. It is thought that shape information included in this parameter can express the lip contour movement, and texture information can express intensity changes such as tooth. Therefore, in this paper, we propose a method that constructs visual HMM using **c** parameter and integrates it with audio HMM. AdaBoost method[9] is employed that uses the Haar-like feature as a face area extraction method, and the late integration that does not take care of audio-visual asynchrony is employed as an integrated method of audio and visual information.

2 System Flow

Fig. 1 shows the block diagram of a processing flow. First, the face area is detected by AdaBoost method that uses the Haar-like feature on the input movie. This is because the extraction accuracy of the feature points by AAM search greatly depends on the initial search area. Therefore, the extraction accuracy of the feature points improves by giving the face area detected by AdaBoost as an initial search area of AAM.

Next, AAM is applied to the detected face area. This process contains two kinds of AAMs. One is the whole face AAM constructed with the training image set in which the feature points are given manually beforehand. The other is the lip area AAM constructed with feature points of the lip area.The purpose of utilizing two AAMs is to extract the feature points accurately on the lip area by applying the whole face AAM roughly at first and then applying the lip area AAM precisely on the extracted lip area. If **c** parameter extracted from the whole face AAM is used as a recognition parameter, the recognition rate might decrease by the information other than the lip area. Therefore, we use two kinds of AAMs to extract a more accurate parameter of the lip area.

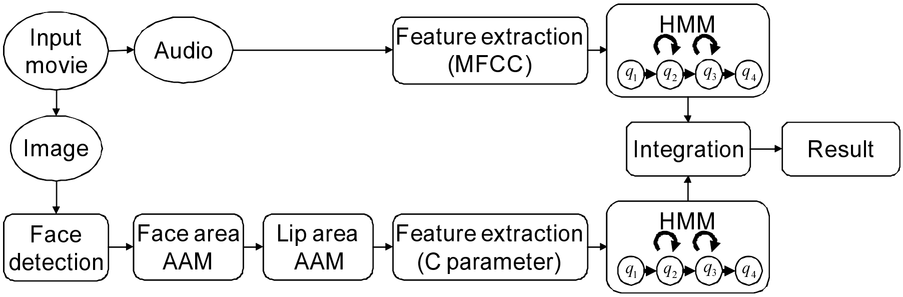


Fig. 1. System Flow

When the lip area AAM is applied to the input image, $\mathbf{c}$ parameter that generates the most similar lip area image with the input image is extracted as the visual feature. In training, audio and visual HMMs are independently constructed by using the visual feature and audio feature extracted from the same movie. Finally, the recognition result is output by integrating likelihoods from visual HMM and audio HMM.

3 Feature Extraction

3.1 Active Appearance Models

AAM is a technique to express the face model by the low-dimensional parameter. The subspace is constructed by applying PCA to shape and texture of face feature points.

The shape vector $\mathbf{s}$ that is composed of the feature points on the face image and mean shape $\bar{\mathbf{s}}$ is computed from the training image set. Inner texture of $\mathbf{s}$ is normalized to mean shape. The shape vector $\mathbf{s}$ and the texture vector $\mathbf{g}$ are given in $\mathbf{s} = (x_1, y_1, \cdots, x_n, y_n)^T$, $\mathbf{g} = (g_1, \cdots, g_m)^T$. where x_i, y_i $(1 \leq i \leq n)$ are the coodinates of the feature points. g_j $(1 \leq j \leq m)$ is the intensity value at each pixel within the area srrounded by $\bar{\mathbf{s}}$, and mean intensity value $\bar{\mathbf{g}}$ can be computed from the training image set. Vectors $\mathbf{s}$ and $\mathbf{g}$ are expressed by using eigenvector matrices $\mathbf{P_s}$ and $\mathbf{P_g}$, obtained by applying PCA to deviation from $\bar{\mathbf{s}}$ and $\bar{\mathbf{g}}$, as shown in Eq. (1) and Eq. (2).

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P_s b_s} \tag{1}$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P_g b_g} \tag{2}$$

$\mathbf{b_s}$ and $\mathbf{b_g}$ are called the shape parameter and the texture parameter respectively, and shape vector $\mathbf{s}$ and texture vector $\mathbf{g}$ are converted to them. Moreover, $\mathbf{b_s}$ and $\mathbf{b_g}$ are combined and reduced as shown in Eq. (3) by applying PCA because there is a correlation in shape and texture parameters.

$$\mathbf{b} = \begin{pmatrix} \mathbf{W_s b_s} \\ \mathbf{b_g} \end{pmatrix} = \begin{pmatrix} \mathbf{W_s P_s^{\mathit{T}}(s - \bar{s})} \\ \mathbf{P_g^{\mathit{T}}(g - \bar{g})} \end{pmatrix} = \begin{pmatrix} \mathbf{Q_s} \\ \mathbf{Q_g} \end{pmatrix} \mathbf{c} = \mathbf{Qc} \tag{3}$$

where $\mathbf{W_s}$ is the matrix that normalizes the difference of the unit between the shape vector and the texture vector. $\mathbf{Q}$ is an eigenvector matrix, and $\mathbf{c}$, called combined parameter, is a parameter that controls both shape and texture. $\mathbf{s}$ and $\mathbf{g}$ are expressed as shown in Eq. (4) and Eq. (5) by $\mathbf{c}$.

$$\mathbf{s(c)} = \bar{\mathbf{s}} + \mathbf{P_s W_s}^{-1}\mathbf{Q_s c} \tag{4}$$

$$\mathbf{g(c)} = \bar{\mathbf{g}} + \mathbf{P_g Q_g c} \tag{5}$$

Thus, it becomes possible to treat shape and texture together by controlling parameter vector c.

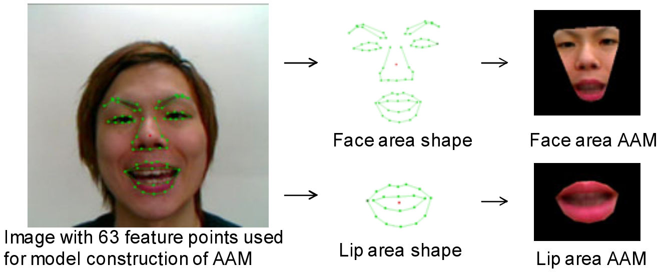


Image with 63 feature points used for model construction of AAM
Face area shape
Face area AAM
Lip area shape
Lip area AAM

Fig. 2. Construction of two kinds of AAMs

3.2 Model Construction

Two kinds of AAMs are constructed as described in Chapter 2. The whole face AAM is constructed by using the shape information and the inside texture information from the training image set with the feature points manually given to the whole face as shown in Fig. 2. The lip area AAM is constructed with the shape information and the inside texture information extracted automatically from the feature points only on the lip area extracted by the whole face AAM.

3.3 Combined Parameter

Since the images with the mouth opening and closing are included in the training data set of AAM, the various movements of the lip can be expressed by changing **c** parameter as shown in Fig. 3. Since **c** parameter has information on detailed shape and the intensity value of the lip, we propose to utilize **c** parameter as the visual feature. As an extraction method of **c** parameter, error **e** between the image $\mathbf{g}(\mathbf{c})$ generated by AAM (this is called a model image) and the input image is formulated as shown in Eq. (6).

$$\mathbf{e}(\mathbf{c},\ \mathbf{p})\ =\ \|\mathbf{g}(\mathbf{c})\ -\ \mathbf{I_i}(\mathbf{W}(\mathbf{p}))\|^2 \tag{6}$$

where $\mathbf{I_i}(\mathbf{W}(\mathbf{p}))$ is the image obtained by Affine transform to the input image $\mathbf{I_i}$. **p** is an Affine parameter of scaling, rotation and translation and **W** is a function that executes the Affine transform. The number of dimension of **c** is set to 10. 78 training images are prepared. Since the video rame rate is about 1/3 of audio frame rate in our data set, there is a possibility that the visual recognition rate decreases compared to the audio recognition rate. Therefore, it is interpolated by the cubic spline function between visual frames. **c** parameters obtained thus, its Δ and $\Delta\Delta$ coefficients with 30 dimensions in total are finally used as the visual feature.

3.4 Additional Feature

In order to compare with **c** parameter, 2D DCT and pixel values on the lip area are extracted. The lip area is located by the whole face AAM, and the area is

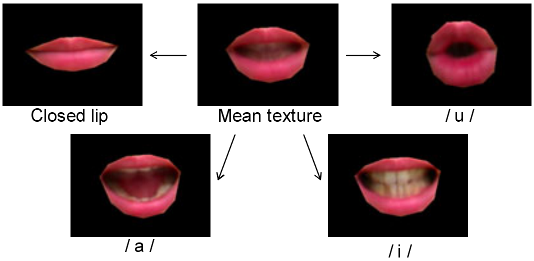


Fig. 3. Example of model images generated by changing c parameter (in a counter-clockwise fashion from the top middle, mean texture, the closed lip, utterance /a/, /i/ and /u/.)

normalized to the square with the fixed ratio of width to hight and converted into the gray scale. The feature is extracted on this area. A square size is 32 32 pixel. PCA is applied to this 1024 dimensional vector of pixel values for the dimension reduction. The number of dimension is set to 10 according to the cumulative contribution ratio 90% . PCA score, its Δ and $\Delta\Delta$ coefficients with 30 dimensions in total are used as the feature of PCA score. In a case of 2D DCT, after DCT operation, 16 low-frequency components are selected because the information concentrates on the low-frequency region in DCT. DCT, its Δ and $\Delta\Delta$ coefficients with 48 dimensions in total are used as the feature of DCT.

4 Recognition Method

As a recognition method, both word type HMM and subword type HMM are used. MFCC with 12 dimensions and logarithm power, their Δ and $\Delta\Delta$ coefficients with 39 dimensions in total are used as the audio feature. A final likelihood is calculated by the late integration of audio and visual information as shown in Eq. (7)[2].

$$L_{A+V} \;=\; \alpha L_A \;+\; (1-\alpha)L_V \;, \quad 0 \leq \alpha \leq 1 \tag{7}$$

where L_{A+V} is a likelihood after integration, L_A and L_V are likelihoods of audio and visual features respectively. α is the combination weight.

5 Experiment

5.1 Experimental Condition

We used ATR phoneme balance words (216 words)×10 sets and single set of 100 words (different from 216 words) chosen at random from ATR phoneme balance sentences as an utterance words. Logicool Qcam Orbit MP was used as a filming equipment and SONY ECM-PC50 was also used as a microphone. Resolution was 960×720 pixel, and the frame rate was 30fps.

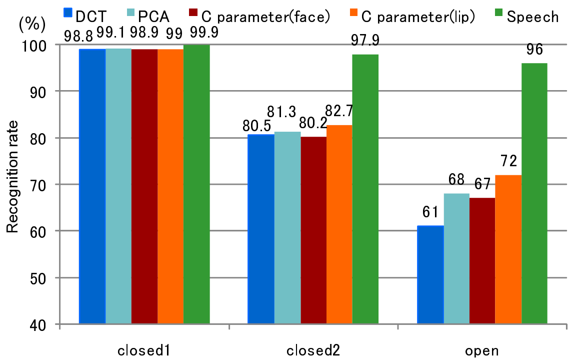


Fig. 4. Recognition results by various audio and visual features in different conditions

One specific speaker uttered in a clear tone with the frontal face. The distance from the speaker to the camera was about 40cm. The noise was added onto the speech so that SNR became 5dB, 0dB and -5dB. The leave-one-out method was applied to 216 words×10 sets, and the recognition rate was the average over the 10 sets. We call this experiment as one under the language closed condition because the same 216 words are used for training and recognition. In addition, 216 words×10 sets were used for training, and 100 words×1 set were recognized. We call this experiment as one under the language open condition, because 100 words are recognized different form 216 words used for training. Word type HMMs were constructed with 5 states and 4 mixtures and used in the language closed condition. As subword type HMMs, monophone HMMs were constructed and used in both the language closed and open conditions. The number of mixture was experimentally chosen for the best one in the language open condition.

5.2 Recognition Result by Using Respective Feature

Fig. 4 shows the result of the utterance recognition carried out separately using the visual feature and audio feature respectively. Closed1 in Fig. 4 indicates the recognition rate by word type HMM, closed2 is by subword type HMM in the language closed condition, and open is in the language open condition. C parameter(face) and C parameter(lip) indicate the recognition results by **c** parameter extracted from the whole face AAM and the lip AAM respectively.

Comparing these results in terms of the features, a high recognition rate was obtained by the conventional features and **c** parameter in closed1. Moreover, it was confirmed that the lip area **c** parameter was more effective than the conventional features in closed2 and open.

Comparing these results in terms of the conditions, the recognition rate decreased in closed2 and open compared with closed1 for the visual feature while it was high in any condition for audio feature. The difference of the conditions between closed1 and closed2 was the HMM type; word type HMM or subword

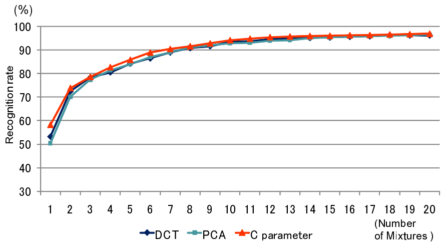


Fig. 5. Recognition rates as a function of the number of mixtures(closed2)

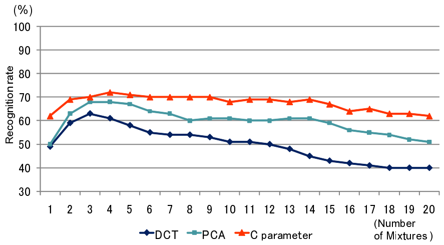


Fig. 6. Recognition rates as a function of the number of mixtures(open)

type HMM. The recognition rate by the subword type HMM was lower than that by the word type HMM because connected training of the phoneme was necessary for the subword type HMM. In the open condition, the recognition rate was lower than that in closed2. Fig. 5 and 6 show the recognition rates by the visual HMMs as a function of the number of mixtures. In the figure, as the number of mixtures increases, the recognition rate is improved in closed2. Since the increase of the number of mixtures leads to the complex model and the training words and test words are same in closed2, it seems that the model is over-fitted to the training data. On the other hand, the recognition rate tends to be lower as the number of mixtures increases in open. Due to this reason, in closed2, the recognition rate is higher than that in open.

5.3 Integrated Result of Audio and Visual Features

In order to integrate the visual result with the audio result under noisy environment, output likelihood by visual HMM with **c** parameter and that by audio HMM were integrated by Eq. (7). Fig. 7 shows the recognition results at 5dB, 0dB and -5dB SNR of the speech data. The weight $1 - \alpha$ to visual feature was increased by 0.1 from 0.0 to 1.0.

Three types of integration of the visual HMMs were carried out with the subword type audio HMM. They were word type visual HMM(closed1), subword type visual HMM(closed2) in the language closed condition and subword type visual HMM(open) in the language open condition respectively. A horizontal axis in Fig. 7 indicates the weight to visual feature. The weight 0 corresponds to audio feature only, and 1 to visual feature only.

From Fig. 7, it can be seen that, in any conditions, the recognition rate is comparatively acceptable in clean and 5dB SNR environment. Therefore, the recognition rate is high at any values of the weight and is improved by taking the optimum value of the weight. The recognition rate by audio HMM greatly falls down in the strong noisy environment at 0dB and -5dB SNR. However, it can be improved by increasing the weight to the image. From these results, it can be confirmed that the recognition rate is improved compared with audio feature by integrating the visual feature and audio feature under noisy environment.

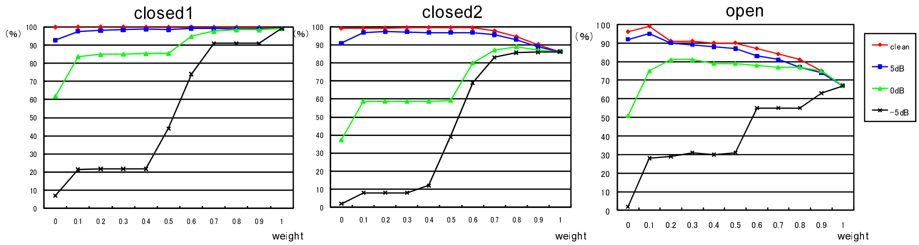


Fig. 7. Integrated result of audio and visual features

6 Phoneme Analysis of Visual Feature

6.1 Continuous Phoneme Recognition

In order to investigate the recognition accuracy of each phoneme using audio and visual features, continuous phoneme recognition was carried out for words. The language model was phoneme pair such that vowel appears after consonant and consonant appears after vowel at equal probability. The acoustic model and the visual model were the subword type audio HMM and the subword type visual HMM trained by 216words×10 sets, and the recognition words were 100 words used in the language open condition. The visual feature was **c** parameter.

Fig. 8 shows the confusion matrix of the phoneme recognition in language open condition by audio features, and Fig. 9 shows the confusion matrix of the phoneme recognition in language open condition by **c** parameter. ”IN” and ”LA” in the figure indicate the number of insertion errors and the number of deletion errors respectively. MoreoverCin order to evaluate the phoneme recognition accuracy, the phoneme correct and the phoneme accuracy of vowel, consonant and all phonemes were computed. The phoneme correct and the phoneme accuracy correspond to word correct and word accuracy respectively when the phoneme is regarded as a word.

Table 1 shows the result. In the table, the recognition accuracy is approximately 80% in both vowel and consonant in audio. However, the recognition accuracy of consonant is about 12% in open condition by visual feature though vowel is approximately 70%, and the accuracy of all phonemes is approximately 40%. Thus, it can be said that consonants are not recognized well by the visual feature.

6.2 Analysis of False Recognition of the Phoneme

In Fig.8 and Fig.9, both vowel and consonant recognition accuracies are high by audio feature. On the other hand, in **c** parameter, vowels are recognized well to some degree, but various errors occur more than audio feature in consonants.

The insertion error occurs a lot in ”r”. It is thought that the shape of the mouth becomes same in ”a” and ”ra” and it can not be distinguished because ”r” is a consonant uttered by the movement of the tongue only. The deletion

Table 1. Phoneme correct and phoneme accuracy (%)

	Audio		Visual			
	Open		Open		Closed2	
	Accuracy	Correct	Accuracy	Correct	Accuracy	Correct
Vowel	82.91	82.91	67.81	68.38	65.46	66.21
Consonant	72.4	75.38	11.85	21.58	37.46	45.87
All	77.65	79.26	40.74	45.74	52.86	57.05

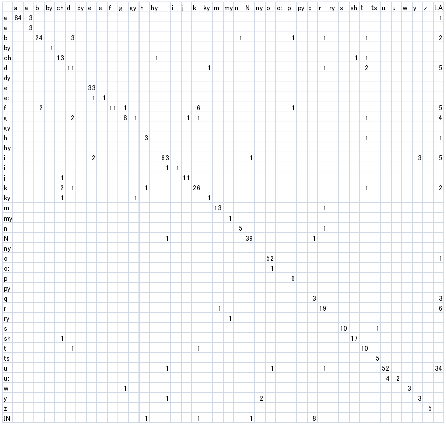


Fig. 8. Phoneme confusion matrix by audio feature (open)

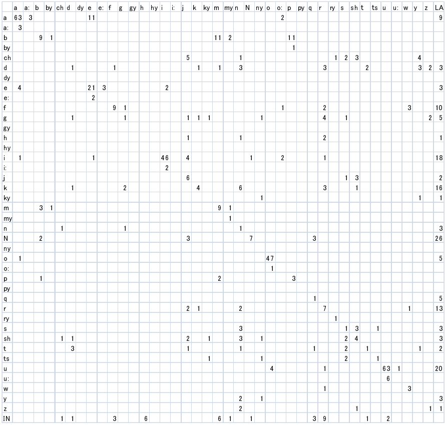


Fig. 9. Phoneme confusion matrix by visual feature (open)

error occurs a lot in "N". When "N" appears at the end of the word, the mouth becomes in a closed shape. Since the mouth is closed before and after the utterance, it is regarded as a silent section, then the deletion error occurs. Moreover, when "N" appears in the word, the shape of the mouth is kept similar to the previous vowel. Therefore, it is thought that the deletion error is increased because "N" has a large variance and sparse feature. The substitution error occurs in various phonemes. For instance, "k" is falsely recognized as the consonants such as "g", "n" and "r". It is thought that the substitution error occurs because there is no movement of the mouth in these consonants.

6.3 Experiment with Viseme

The reason why the false recognition described in 6.2 is caused is attributed to the fact that the phoneme is a minimum unit representing the sound. When the phoneme is applied to the visual feature, the phonemes with the same shape of the mouth such as "k" and "g" cannot be distinguished. Therefore, the viseme will be the best unit, instead of the phoneme, to represent the visual feature.

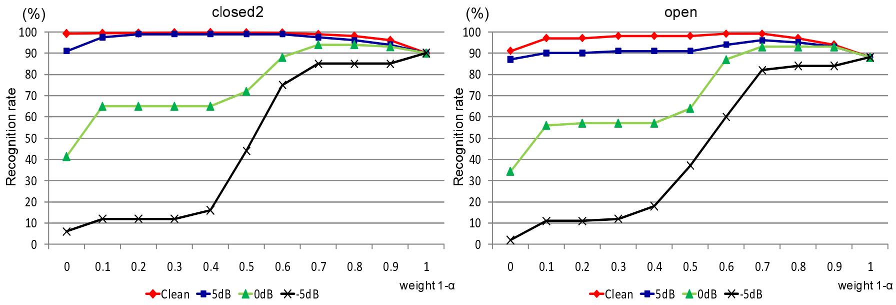


Fig. 10. Integrated result when the viseme is used for visual information and phoneme is used for audio information

Table 2. Viseme correct and viseme accuracy (%)

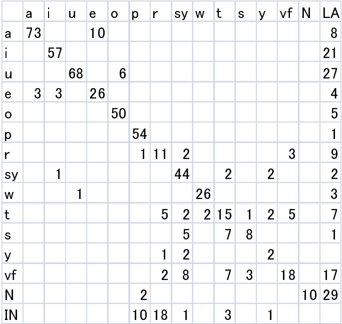
	Open		Closed2	
	Accuracy	Correct	Accuracy	Correct
Vowel	75.9	75.9	78.21	78.58
Consonant	47.69	57.85	63.28	68.44
All	62.54	67.35	71.59	74.08

From this viewpoint, the viseme was employed as a unit to represent the visual feature, referring to Fukuda[10], and the visual data was recognized as was done in Chapter 5 by visual HMM and the result was integrated with the audio result. The number of mixtures was set to 12 based on the best result using the viseme. There were some words that could not be distinguished like "eikyou" and "eigyou" because both became "eisyou" in viseme. For such words, the same output likelihood from the visual HMM was integrated with those from the audio HMMs with different phoneme sequence. Fig. 10 shows the integrated result in closed2 and open.

In the figure, it can be confirmed that the recognition results are better than those in Fig. 7, because the recognition rate by the visual HMM using viseme is higher than that using phoneme shown in Fig. 7. Therefore, the highest accuracy is obtained by integrating the recognition results using phoneme for audio feature and viseme for visual feature.

As the experiment, the continuous viseme recognition was carried out. Fig. 11 shows the confusion matrix, and Table 2 shows the correct and the accuracy when viseme is used.

Comparing Table 2 with Table 1, the viseme greatly improved the recognition accuracy in both vowels and consonants, compared to the phoneme case. However, it is still low by about 10 points in closed2 compared to audio. In Fig. 11, "N" has still many deletion errors as is described in 6.2 for the phoneme, and "t" has many substitution errors with various visemes. Viseme "t" includes the phoneme "t", "d" and "n". In order to discriminate these, it is important



	a	i	u	e	o	p	r	sy	w	t	s	y	vf	N	LA
a	73			10											8
i		57													21
u			68		6										27
e	3	3		26											4
o					50										5
p						54									1
r						1	11	2					3		9
sy		1						44		2		2			2
w			1						26						3
t							5	2	2	15	1	2	5		7
s								5		7	8				1
y							1	2				2			
vf							2	8		7	3		18		17
N						2								10	29
IN						10	18	1		3		1			

Fig. 11. Viseme confusion matrix using c parameter(open)

to extract the movement of the tongue because they are uttered by changing the tonge position. Moreover, if they can be discriminated, the accuracy of the viseme "vf" will be improved that has many substitution error to "t".

It is thought that there will be still room in the improvement of the visual feature. In the future, we will investigate the feature that can be extracted from the movement of the tongue described above, and the feature that can recognize "N" clearly.

7 Conclusion

We proposed to utilize **c** parameter extracted by Active Appearance Model applied to a face image for the utterance recognition. The effectiveness was confirmed by integrating **c** parameters as the visual feature with the audio feaure. The difference between the phoneme recognition accuracy by the audio feature and the visual feature was clarified by calculating the phoneme confusion matrix. In addition, the phoneme score from audio feature and the viseme score from visual feature were integrated with high accuracy.

In our approach, the utterances spoken by one specific speaker with a clear tone were recognized in the experiment. Future tasks include the recognition of utterances spoken by more people, new integration method of audio and visual feature, weight optimization technique, recognition of speech with spontaneous tone, application of AAM to images with various face directions, expansion to continuous speech recognition, and robustness to the difference of time session. Though monophone type HMM was used in this experiment because of the data amount, a further improvement of the recognition rate will be expected by increasing the data amount and using triphone type HMM.

References

1. Potamianos, G., Graf, H.P.: Discriminative Training Of HMM Stream Exponents For Audio-Visual Speech Recognition. In: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1998), Florham Park, NJ, pp. 3733–3736 (1998)

2. Verma, A., Faruquie, T., Neti, C., Basu, S., Senior, A.: Late Integration In Audio-Visual Continuous Speech Recognition. In: Automatic Speech Recognition and Understanding (1999)
3. Tomlinson, M.J., Russell, M.J., Brooke, N.M.: Integrating audio and visual information to provide highly robust speech recognition. In: Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP1996), pp. 821–824 (1996)
4. Kumar, K., Navratil, J., Marcheret, E., Libal, V., Ramaswamy, G., Potamianos, G.: Audio-Visual Speech Synchronization Detection Using a Bimodal Linear Prediction Model. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 53–59 (1999)
5. Iwano, K., Tamura, S., Furui, S.: Bimodal speech recognition using lip movement measured by optical-flow analysis. In: Proc. International Workshop on HSC 2001, pp. 187–190 (2001)
6. Jun, H., Hua, Z.: Research on Visual Speech Feature Extraction. In: 2009 International Conference on Computer Engineering and Technology, pp. 499–502 (2009)
7. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
8. Dornaika, F., Ahlberg, J.: Fast and reliable active appearance model search for 3-d face tracking. IEEE Transactions on Systems, Man, and Cybernetics, 1838–1853 (2004)
9. Viola, P., Jones, M.: Rapid Object Detection Using Boosted Cascade of Simple Features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–9 (2001)
10. Fukuda, Y., Hiki, S.: Characteristic of the mouth shape in the production of Japanese-Stroboscopic observation. In: IEICE, pp. 259–265 (1978)

Multi-scale Integration of Slope Data on an Irregular Mesh

Rafael F.V. Saracchini[1], Jorge Stolfi[1], Helena C.G. Leitão[2],
Gary Atkinson[3], and Melvyn L. Smith[3]

[1] State University of Campinas,
Campinas, Brazil
{ra069320,stolfi}@ic.unicamp.br

[2] Fluminense Federal University,
Niteroi, Brazil
hcgl@ic.uff.br

[3] University of West England,
Bristol, United Kingdom
{Gary.Atkinson,Melvyn.Smith}@uwe.ac.uk

Abstract. We describe a fast and robust gradient integration method that computes scene depths (or heights) from surface gradient (or surface normal) data such as would be obtained by photometric stereo or interferometry. Our method allows for uncertain or missing samples, which are often present in experimentally measured gradient maps; for sharp discontinuities in the scene's depth, e.g. along object silhouette edges; and for irregularly spaced sampling points. To accommodate these features of the problem, we use an original and flexible representation of slope data, the weight-delta mesh. Like other state of the art solutions, our algorithm reduces the problem to a system of linear equations that is solved by Gauss-Seidel iteration with multi-scale acceleration. Its novel key step is a mesh decimation procedure that preserves the connectivity of the initial mesh. Tests with various synthetic and measured gradient data show that our algorithm is as accurate and efficient as the best available integrators for uniformly sampled data. Moreover our algorithm remains accurate and efficient even for large sets of weakly-connected instances of the problem, which cannot be efficiently handled by any existing algorithm.

1 Introduction

The *integration of a gradient map* to yield a height map is a computational problem that arises in several computer vision contexts, such as shape-from-shading [9,8] and multiple-light photometric stereo [10,22]. These methods usually determine the mean surface normal vector within each image pixel, from which one can obtain the height gradient (the partial derivatives of the surface's height Z with respect to the spatial coordinates X and Y). See figure 1.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 109–120, 2011.

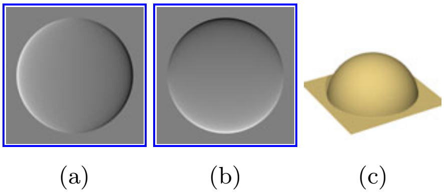


Fig. 1. Derivative maps $\partial Z/\partial X$ and $\partial Z/\partial Y$(b,c) of a hemisphere and the height map(c) obtained by integration

Although this information alone does not determine the absolute surface heights, it can yield height differences between parts of the same surface. This relative height information is sufficient for many important applications, such as industrial quality control [18], pottery fragment reassembly [11], surveillance, face recognition [13], and many others.

In practical contexts, this problem faces at least four difficulties. First, the gradient data is usually *discretized*, that is, given as a finite set of *gradient samples*, each being an average of the gradient ∇Z over some neighborhood of a *gradient sampling point.*

Second, the gradient data is usually contaminated with *noise* arising from unavoidable measurement, quantization, and computation errors.

Third, the height function $Z(X, Y)$ of a real scene is usually *discontinuous*. In particular, it almost always has step-like discontinuities, or *cliffs*, at the edges of solid objects. Most gradient acquisition methods, such as photometric stereo, will return meaningless values for any sample that straddles a cliff or that cannot be measured. For this reason, practical integration algorithms require an additional input, a real-valued *weight map* that specifies the reliability of each gradient sample. The weight map can be just a binary mask that is zero where there is invalid data or cliffs and 1 elsewhere. See figure 2.

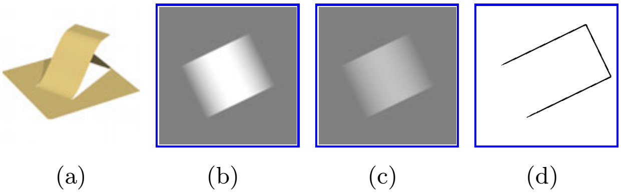


Fig. 2. A height map with cliff-like discontinuities (a), the derivative maps $\partial Z/\partial X$ and $\partial Z/\partial Y$(b,c), as could be obtained by photometric stereo methods, and a binary mask (d) showing the location of the cliffs. Note that the gradient map is oblivious to the cliffs, and gives no clue as to which end of the ramp (if any) is at ground level.

Finally, even if the data is initially acquired over a regular X–Y grid of sampling points, the samples may become irregularly spaced when the data is subjected to optical rectification, filtering, or interpolation.

2 Previous Solutions

There is a substantial bibliography on the gradient-to-height problem of computer vision, beginning with B. K. P. Horn's seminal papers [9,8]. Three surveys have been published by Agrawal [3], Ng *et al.* [14], and Saracchini *et al.* [16]. The published solution methods fall into a few major classes:

Path integration methods [5,15,2] compute the relative height of each pixel as a line integral along a single path from some reference pixel. These methods are very efficient ($\Theta(N)$ time and space, where N is the number of data pixels), but are extremely sensisitive to noise present in the gradient data and generally yield height maps with spurious cliffs. See figure 3.

Spectral methods, such as those of Frankot-Chellappa [6], Georghiades [7], and Wei [21] use the fast Fourier transform (FFT) to perform the integration by filtering the gradient data in the frequency domain. These methods are only slightly more expensive than path integration ($\Theta(N \log N)$ time and $\Theta(N)$ space) and fairly immune to random data noise. However, they cannot handle data with cliffs or missing samples, since the FFT only works with regularly spaced data and gives the same weight to every sample. When applied to scenes with cliffs, these methods return severely distorted height maps. See figure 4.

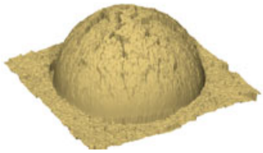


Fig. 3. Output of Fraile-Hancock's integrator [5] applied to the gradient data of figure 1 with noise added

Kernel methods, introduced by Ng *et al.* [14] assume a sparse gradient field, and reduce the problem to data fitting with a high-dimensional function approximation space. This approach can accomodate irregularly spaced gradient sampling points and is claimed to provides better "fill in" for missing data than Poisson methods. However it requires solving a very large ($3N \times 3N$) linear equation system, and is therefore way expensive in time and space.

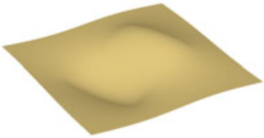


Fig. 4. Output of the Frankot-Chellappa integrator [6] applied to the gradient data of figure 2

Direct Poisson-like methods reduce the problem to an $N \times N$ sparse system of equations which is solved directly through Gaussian or Cholesky factorization, as described by Agrawal [3]. The system can be obtained in many equivalent ways, such as by analogy to the Poisson second-order differential equation [3], through an energy minimization formulation [3], as the least squares solution to an overdetermined system [8], or by a local averaging principle [17]. These methods can take into account weight maps, modify the Poisson system so as to use only valid data and avoid integrating around cliffs. As result, they can handle problems that path-based and spectral methods cannot.

On the other hand, direct Poisson-like methods can be quite expensive. The solution of the system requires approximately $\Theta(N^{1.5})$ time and $\Theta(N^{1.15})$ space, with large constants factors. The high memory cost makes this approach impractical for megapixel gradient maps [16].

Iterative Poisson-like methods build the same linear system as the direct variant, but solve it by the iterative Gauss-Seidel method [19]. With this approach the memory space needed is only $\Theta(N)$, but the time to achieve a preset accuracy grows at least proportionally to N^2; so that even modest (100×100) gradient maps may require more than 10^5 iterations to produce a minimally usable result.

Multi-scale Poisson-like methods, first described by Terzopoulos in 1986 [20,19], use *multi-scale techniques* to acelerate the Gauss-Seidel iterative algorithm. The idea is to recursively solve a coarse version of the original problem, with the gradient maps reduced to half size; and then use the resulting heigh map, expanded back to the original scale, as the initial guess for the Gauss-Seidel iterator.

Let $\varepsilon^{(k)}$ be the residual error, namely the difference between the current guess and the true solution, after k Gauss-Seidel iterations. As observed by Terzopoulos [19], the slow convergence of the Gauss-Seidel method is due to the Fourier components of $\varepsilon^{(k)}$ with low spatial frequency, which decrease very little at each iteration. The high-frequency components of the error, on the other hand, are quickly eliminated after a few iterations. Thus, the recursively computed initial guess will provide the correct low-fequency components of the solution, and the Gauss-Seidel loop quickly fixes the high frequency components. A fast weighted Poisson-based integrator along these principles was developed in by Saracchini *et al.* [16].

2.1 The Problem of Weakly Connected Data

The multi-scale approach fails when the slope maps contain narrow bands of data surrounded by cliffs or missing samples. When the weight map is reduced, any pixel of the result that contains a zero weight pixel of the original must be set to zero too, since it may contain a cliff. It follows that the relative area affected by the missing samples expands at each successive reduction, until the narrow bands of data disappear and/or the connectivity of the gradient map is broken. See figure 5.

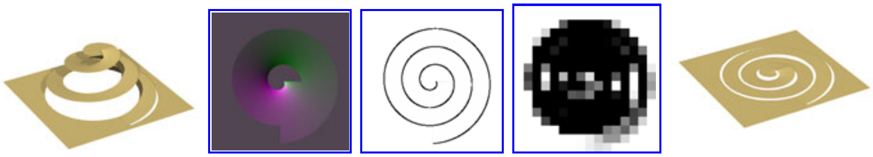


Fig. 5. A height map, its gradient map, weight map (256x256 and 16x16 scale) and the integrator's output [16] after 200 iterations

At that point, the solution computed for the reduced problem is no longer a suitable starting guess, since its low-frequency components are usually quite wrong. On such maps, the multiscale Gauss-Seidel solver becomes considerabily slower than the direct Gauss or Cholesky solver.

3 Integration on an Irregular Mesh

Our algorithm is a Poisson method with a novel multiscale iterative solver, that is effective even for weakly-connected instances like that of figure 5.

The Weight-Delta Mesh Model. We depart from tradition by using a graph representation for the gradient and weight data, instead a regular grid of samples. A *weight-delta mesh* (WDM) is an abstract directed planar graph G with vertices (nodes) $\mathcal{V}\,G$ and edges (arcs) $\mathcal{E}\,G$. Each vertex v represents a height sampling point and is associated to an unknown height value $z[v]$. Each directed edge e connects two close vertices and has two numeric parameters: the *edge delta* $d[e]$, and the *edge weight* $w[e]$.

The edge delta $d[e]$ is an estimate for the difference $z[v] - z[u]$ between the height values at the edge's origin vertex $u = \textsc{org}(e)$ and its destination vertex $v = \textsc{dst}(e)$. This estimate is presumably derived from measured surface gradients between the corresponding height sampling points; the details of this computation depend on the application and are not relevant to this paper. The edge weight $w[e]$ is a positive number that expresses the reliability of that estimate. More precisely, we assume that the edge delta $d[e]$ includes some Gaussian measurement error (provenient from camera noise, quantization,etc.), whose expected value is zero and whose variance is proportional to $1/w[e]$.

By definition, a weight-delta mesh has no loop edges. We say that a WDM is *simple* if it is free from parallel edges (two or more edges with same origin and destination). In a simple WDM, we can identify each edge e with the ordered pair (u, v) of its origin and destination vertices. In that case we may denote $d[e]$ also by $d[u, v]$, and $w[e]$ by $w[u, v]$. Also by definition, for every directed edge e in a WDM, the oppositely directed edge $\textsc{sym}(e)$ is also present in the mesh, with $d[\textsc{sym}(e)] = -d[e]$ and $w[\textsc{sym}(e)] = w[e]$. Therefore, when drawing the mesh it suffices to draw only one directed edge out of each pair $e, \textsc{sym}(e)$. See figure 6.

Edge Equations. A WDM can be interpreted as an equation system, with one *edge equation*

$$z[\textsc{dst}(e)] - z[\textsc{org}(e)] = d[e] \tag{1}$$

for every directed edge e. This equation is assumed to have "strength" $w[e]$. The problem is then to solve this system for the height $z[v]$ of each vertex v, given the mesh and the parameters $d[e], w[e]$ for every graph edge e.

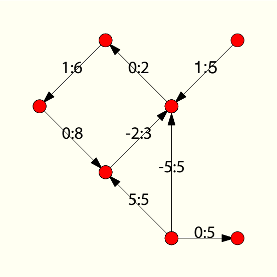


Fig. 6. A small WDM. The edge labels are $d[e]$:$w[e]$.

Since each connected component of the WDM implies a separate set of unknowns and equations, we will henceforth assume that the WDM is a connected graph. Note that the edge equations (1) only depend on height differences; therefore the solution for a connected mesh has at least one degree of freedom (an additive term corresponding to the integration constant of the continuous problem).

Vertex Equilibrium Equations. If G has cycles, the edge equation system (1) is overdetermined. In that case, measurement errors present in the deltas often make it impossible to satisfy all equations at the same time. Given the assumption of independent Gaussian measurement errors in the d values, Bayesian analysis says that the most likely set of heights z is the weighted least squares solution to the system (1). That solution turns out to satisfy the *vertex equilibrium equation*

$$z[u] - \sum_{v \in G[u]} \lambda[v] z[v] = - \sum_{v \in G[u]} \lambda[v] d[u,v] \tag{2}$$

for every vertex u, where $G[u]$is the set of vertices adjacent to u in the mesh and $\lambda[v]$ is $w[u,v]/\sum_s w[u,s]$, the *relative* weight of v among the neighbors of u.

4 The Algorithm

The core of the algorithm is a *mesh decimation* step that removes a certain fraction of the vertices of the input mesh G, producing a smaller mesh G'. The vertices of G' are a subset of those of G, and the edges of G' are defined so as to best summarize the weight and delta information contained in the edges of G. The algorithm then solves the problem recursively for the mesh G' yielding a tentative height function z' for its vertices. It then interpolates heights to provide a starting guess z for the original mesh G. Finally it adjusts the heights z by applying few Gauss-Seidel iterations to the equilibrium equations (2).

The recursion stops when G is reduced to a single vertex v, whose height $z[v]$ can be set to zero. In other words, we construct a pyramid $G^{(0)}, G^{(1)}, \ldots, G^{(m)}$ of meshes, where $G^{(0)}$ is the input mesh G, $G^{(m)}$ is a single vertex v, and each mesh $G^{(k+1)}$ is obtained by decimation of the previous one $G^{(k)}$. Then we compute solutions $z^{(m)}, z^{(m-1)}, \ldots, z^{(0)}$, in that order; where $z^{(m)}[v]$ is zero for its single vertex v, and each $z^{(k)}$ is obtained from $z^{(k+1)}$ by mesh interpolation and Gauss-Seidel iteration. The map $z^{(0)}$ is the result. See figure 7.

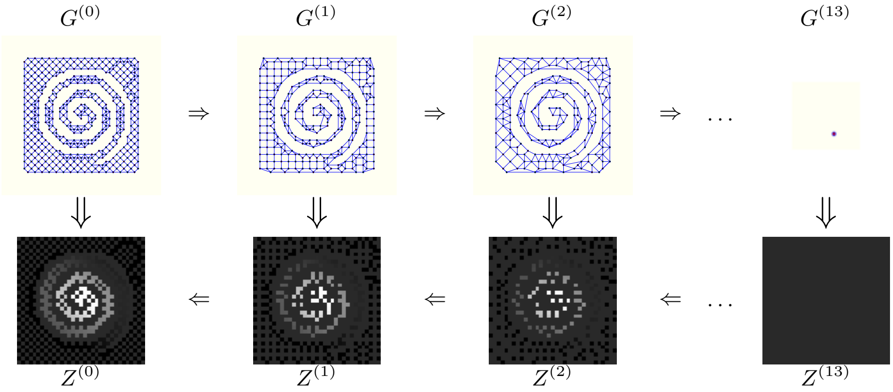


Fig. 7. The multiscale integration method

Formally, the algorithm is the recursive procedure *Integrate* whose pseudocode is given in figure 8. It takes as inputs the weight-delta mesh G, an iteration limit κ and a tolerance ε; and outputs a height function z from $\mathcal{V}\,G$ to $\mathbb{R}$.

```
Integrate$(G, \kappa, \varepsilon)$
  1. If $\#\mathcal{V}\,G = 1$ then
       2. Let $v$ be the only vertex in $\mathcal{V}\,G$; set $z[v] \leftarrow 0$;
  3. else
       4. $G' \leftarrow Decimate(G)$;
       5. $\beta \leftarrow \#\mathcal{V}\,G' / \#\mathcal{V}\,G$;
       6. $z' \leftarrow Integrate(G', \kappa/\sqrt{\beta}, \varepsilon\sqrt{\beta}, )$;
       7. $z \leftarrow Interpolate(z', G', G)$;
       8. $z \leftarrow SolveSystem(z, G, \kappa, \varepsilon)$;
  9. Return $z$.
```

Fig. 8. The main procedure of the integrator

Mesh Decimation. The procedure *Decimate*, called in step 4, takes a simple mesh G, planar and connected, and outputs a smaller mesh G', which is also simple, planar, and connected.

First, the procedure partitions $\mathcal{V}\,G$ into a set R of vertices to be removed, and a set K of vertices to be kept. The set R is a maximal subset of $\mathcal{V}\,G$ whose elements are independent (that is, pairwise disconnected in G) and have degree six or less. The set R is found by a greedy algorithm [4].

Next, the vertices in the R set are removed from G. Whenever a vertex u is removed, the edges incident to u are removed, too. If u has degree 1, nothing else needs to be done. If u has degree 2 or more, new edges are added to G', connecting the neighbors of u. (Observe that all these neighbors are in K and therefore they will be vertices of G'.) The endpoints, weights and deltas of the new edges are chosen so that the solution $z'[v]$ for the mesh G' is as close as possible to the solution $z[v]$, on every vertex $v \in K$.

More precisely, let k be the degree of u in G; let $e_0, e_1, \ldots, e_{k-1}$ be the edges incident to u, oriented out from u, in counterclockwise order around u; and let $v_0, v_1, \ldots, v_{k-1}$ be the corresponding destination vertices. Let w_i be the weight of e_i, and d_i its delta. It can be shown that the solution z' for G' would exactly match the solution z for G if, for every pair i, j, we added an edge $e'_{i,j}$ from v_i to v_j with delta $d'_{ij} = d_j - d_i$ and weight $w'_{ij} = w_i w_j / w_{\mathrm{tot}}$, where w_{tot} is the sum of all weights w_i. We call this operation — removal of u, removal of all incident edges e_i, and the addition of all edges e'_{ij} — a *star-clique swap.*

If the vertex has degree $k = 2$, the swap will add only one pair of opposite edges e'_{01} and e'_{10}. If the degree k is 3, there will be three new edge pairs: e'_{01}, e'_{12}, e'_{02}, and their opposites. In both cases, the planarity of the mesh G is preserved. However, when the degree k is 4 or more, adding all the $k(k-1)$ directed edges $e'_{i,j}$ would generally make G' non-planar, and would severely impact the algorithm's efficiency.

Therefore, when $k \geq 4$ we use instead a *star-cycle swap*, which adds only the edges $e'_{i,i+1}$ that connects successive vertices v_i and v_{i+1}, for $i \in \{0, 1 \ldots k-1\}$ into a cycle; as well their opposites. (All indices are taken modulo k). The deltas $d'_{i,i+1}$ of these edges are those of the star-clique swap, namely $d'_{i,i+1} = d_{i+1} - d_i$. The weights $w'_{i,i+1}$, on the other hand, are given by different formulas for each degree k. For the new edge $e'_{01} = (v_0, v_1)$, we have

k	w'_{01}
2	$w_0 w_1 / w_{\text{tot}}$
3	$(w_0 w_1 + 0.5(w_0 w_2 + w_1 w_3))/w_{\text{tot}}$
4	$(w_0 w_1 + 0.5(w_0 w_2 + w_1 w_3))/w_{\text{tot}}$
5	$(w_0 w_1 + 1.1690(w_2 w_4 + w_0 w_2 + w_1 w_4))/w_{\text{tot}}$
6	$(w_0 w_1 + 2 w_5 w_2 + 1.5(w_5 w_1 + w_0 w_2))/w_{\text{tot}}$

The same formulas hold for any other edge $e'_{i,i+1}$ of the cycle, except that all indices are incremented by i modulo k.

Unlike the star-clique swap, the star-cycle swap does not ensure that the heights determined by G' are exactly equal to those implied by G. However, the solution z' for the mesh G' retains the "low-frequency" components of the solution z of G — in the sense that the error is highly localized, and can be removed by only a few Gauss-Seidel iterations.

An edge e'_{ij} introduced by the star-cycle swap may have the same endpoints as a preexisting edge. Therefore, after performing all the star-cycle swaps, the *Decimate* procedure collapses every set of parallel edges into a single equivalent edge in a way that preserves the final solution. Namely, if edges e' and e'' have the same origin and destination, with weights w', w'' and deltas d', d'', they are replaced by a single edge e with the same endpoints, with attributes

$$w[e] = w' + w'' \qquad d[e] = (w'd' + w''d'')/(w' + w'') \tag{3}$$

Interpolation. Once a solution z' has been obtained for the reduced mesh G' (step 6), it is expanded to a starting guess z for G, by the procedure *Interpolate* (step 7). First, for every vertex v in the shared set K, we set $z[v] \leftarrow z'[v]$. Then, for every vertex u in the deleted set R, we compute $z[u]$ by its vertex equilibrium equation (2). Note that every neighbor $v \in G[u]$ belongs to K, and therefore its height $z[v]$ is defined at this point.

Iterative Adjustment. The initial guess z is then used as the starting guess for the Gauss-Seidel procedure *SolveSystem* (step 8). Each iteration of the latter scans every vertex $u \in \mathcal{V} G$ and uses the equilibrium equation (2) to recompute its height $z[u]$ from the current heights $z[v]$ of its neighbors. The procedure terminates after a specified maximum number κ of iterations, or after the maximum absolute change in any height $z[u]$ is less than the specified tolerance ε, whichever happens first. Note that the iteration limit κ is increased by a factor $1/\sqrt{\beta}$, and the tolerance ε is reduced by $\sqrt{\beta}$, at each level of the recursion (step 6); where β is the mesh size reduction factor achieved by *Decimate* (step 4).

5 Analysis of the Algorithm

Correctness. The star-cycle transformation and the collapsing of parallel edges preserve both planarity and connectivity, so the recursive calls to *Integrate* satisfy its preconditions that G' be simple, connected and planar. Therefore, the connectivity and planarity of the original mesh is preserved at all levels of the pyramid; even within narrow corridors the relevant gradient information is retained all the way to the top. Moreover, if κ is large enough, the final application of the Gauss-Seidel algorithm (at scale 0) will eventually converge to the unique solution $z = z^{(0)}$ of the vertex equations (2), irrespective of the starting guess obtained from the decimated mesh $G^{(1)}$. The experimental tests (section 6) show that convergence is achieved after only a few iterations, even in instances that cause other multiscale methods to fail.

Space and Time Costs. Let $N = \#\mathcal{V}G, N_k = \#\mathcal{V}G^{(k)}$, $M = \#\mathcal{E}G, M_k = \#\mathcal{E}G^{(k)}$. It is known that, for planar simple graphs, $M \leq 6N$ and $M_k \leq 6N_k$; and that any such graph has at least $N/7$ vertices with degree 6 or less. From these facts it follows that the vertex reduction factor β of the *Decimate* procedure has a theoretical upper bound $\hat{\beta} \leq 41/42 \approx 0.976$ [12]. In practice, the reduction factor β is usually 0.6.

The maximum scale m is therefore at most $\log_{1/\hat{\beta}} N = O(\log N)$. Moreover, the total vertex count in all meshes is at most $N/(1-\hat{\beta}) = O(N) \approx 2.5N$ in practice. The amount of memory required by the algorithm is dominated by the representation of the mesh $G^{(k)}$; a simple representation that is sufficient for our purposes uses only $N_k + 2 \times 3M_k \leq 19N_k$ words for the mesh $G^{(k)}$, and $(19/(1-\hat{\beta}))N$ words for all meshes in the pyramid.

The decimation algorithm runs in time $O(N+M) = O(N)$ for a planar graph, therefore the whole pyramid is built in $O(N/(1-\hat{\beta})) = O(N)$ time. The time required for one Gauss-Seidel iteration at level k is $\Theta(N_k + 2M_k) = \Theta(N_k)$. The maximum number of iterations at that level is $q_k = q/\hat{\beta}^{k/2}$. The maximum time spent at level k is then proportional to $N_k q_k = (N\hat{\beta}^k)(q/\hat{\beta}^{k/2}) = N\hat{\beta}^{k/2}$. Therefore, the total work at all levels is $O(N/(1-\hat{\beta}^{1/2})) = O(N)$.

6 Tests

In this section we experimentally compare the cost and accuracy of our graph-based multiscale integrator (MG) with those of other published methods. We consider only weighted Poisson-based algorithms since they are the ones that can cope with errors and discontinuities in the gradient data within an acceptable execution time. Another methods such were not tested due being unable to cope with discontinuities [6], high sensibility towards gradient noise [5] or too high memory/time requirements to be comparable [14].

Specifically, we used the M-Estimators (ME) and Affine Transforms (AT) algorithms [3] of Agrawal *et al.* with direct system solving; and the the multiscale iterative integrator (MS) of Saracchini *et al.* [16]. For ME and AT we used the author's Matlab implementations [1] under MS Windows, adapted to use our input and output file formats and a user-given (rather than internally computed) weight map. For MS we used the author's implementation in C. Our algorithm MG was also implemented in C; both were compiled and tested on a GNU Linux platform. The maximum number of Gauss-Seidel iterations κ was set to 200 for MS (as proposed by the authors) and to 20 for our method.

Datasets. In our tests we used four datasets provided by Saracchini *et al.*, as shown in figure 9. Three of them (`spdome`, `cbabel`, and `cpiece`) are defined by mathematical functions, and one (`dtbust`) is a terrain model of a human torso obtained by a structured-light 3D scanner. In order to simulate the measurement noise usually present in real datasets, we added to each gradient sample an independent Gaussian random number with zero mean and deviation 0.3. Each gradient and its weight map were converted to a WDM whose vertices were the pixels of desired height map and whose arcs connected pixels that were vertically or horizontally adjacent in that map. The final vertex height z were then output in the regular grid format.

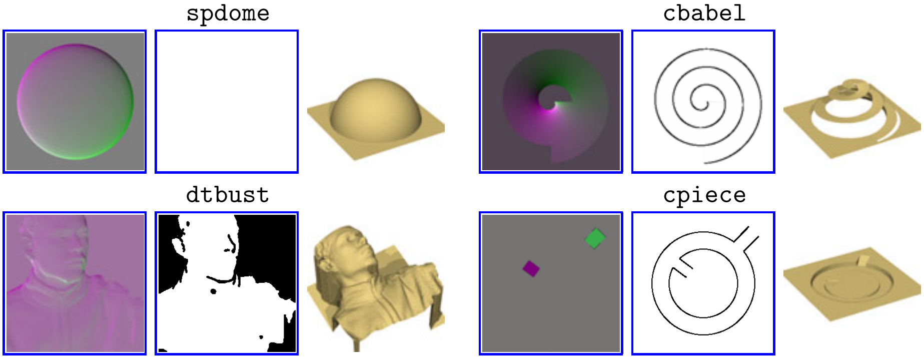


Fig. 9. Datasets used in the tests, showing the gradient maps (left), the weight masks (middle), and the correct height map (right)

Accuracy and Robustness. For each combination of dataset and algorithm, we computed the RMS value ρ of the correct and integrated height fields, and the RMS difference η between them. In these computations, the height fields were first shifted to have zero mean, and all averages are weighted by the input weight maps.

Table 1. Relative RMS errors of each method

Results - datasets with 30% of Gaussian noise								
	`spdome`		`cbabel`		`dtbust`		`cpiece`	
Meth.	η	η/ρ	η	η/ρ	η	η/ρ	η	η/ρ
AT	3.32	9.8%	0.80	3.0%	1.22	4.9%	0.52	4.1%
ME	0.63	1.8%	0.86	3.3%	0.71	2.8%	0.55	4.3%
MS	0.34	1.0%	23.02	121.0%	0.67	2.7%	5.74	52.5%
MG	0.34	1.0%	0.80	3.1%	0.59	2.3%	0.52	4.1%

As table 1 shows in these tests the accuracy of our MG method was equivalent or better than that of the other three. Note that MS integrator failed on the `cbabel` and `cpiece` datasets, due to loss of connectivity after the first few levels of the pyramid. On the `dtbust` dataset, MS gives the correct solution but only after 200 iterations at the base level.

Cost. To evaluate the efficiency of our method, we measured the computing time and memory needed for the integration of two gradient fields sampled with various grid sizes from 64 × 64 to 512 × 512. We used the two datasets which where correctly integrated by all methods (`spdome` and `dtbust`), without noise.

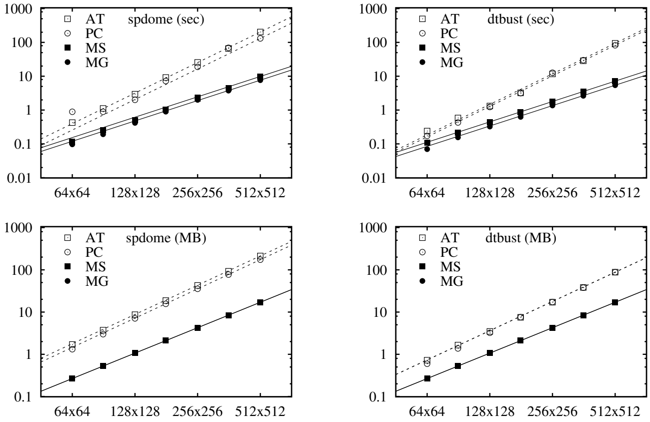


Fig. 10. Log-log plots of the running time (top) and memory usage (bottom) of PC, AT,MS and MG

For AT and ME, we measured only the system solving step; namely, we aborted the algorithm after a single iteration of its weight-computing step. For MS and MG, we included the cost of their decimation/interpolation steps as well as of the Gauss-Seidel solver. The direct solving methods AT and ME need to store the Poisson system's matrix A and also its Gaussian triangular factor U (or Cholesky's R). For those methods, we counted the nonzero entries N_A in the system's matrix A and N_U in its Gauss or Cholesky's factor U, and estimated the memory usage conservatively as $12N_A + 16N_U$ bytes For MS we used the memory estimate given by the authors [16]. For our method we used the estimate $19N_{tot}$ where N_{tot} was the actual number of vertices in all meshes.

The running times of MS and MG cannot be compared directly to those of AT and ME, since Matlab code is inherently slower than C code. However, figure 10 shows that memory and time costs of MS and MG scale linearly with N, where as those of AT and ME scale as $O(N^{1.15})$ and $O(N^{1.5})$, respectively.

7 Conclusions

Our algorithm allows robust integration of slope maps with cliffs and missing data. Unlike previous linear-cost algorithms, it can handle gradient maps with narrow corridors. Also it can be used as the inner loop of iterative methods such as described in [3], were the the computed heights are used to determine the weights of the next iteration, allowing the detection of outliers and noisy data.

References

1. Agrawal, A.: Matlab/Octave code for [3] (2006), `http://www.umiacs.umd.edu/~aagrawal/software.html`
2. Agrawal, A., Chellappa, R., Raskar, R.: An algebraic approach to surface reconstruction from gradient fields. In: Proc. 7th ICCV, pp. 174–181 (2005)
3. Agrawal, A., Raskar, R., Chellappa, R.: What is the Range of Surface Reconstructions From a Gradient Field? In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 578–591. Springer, Heidelberg (2006)
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. McGraw-Hill (1990)
5. Fraile, R., Hancock, E.R.: Combinatorial surface integration. In: Proc. 18th ICPR 2006, vol. 1, pp. 59–62 (2006)
6. Frankot, R.T., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. IEEE TPAMI 10(4), 439–451 (1988)
7. Georghiades, Belhumeur, Kriegman: Illumination cone models for face recognition under variable lighting and pose. IEEE TPAMI 23, 643–660 (2001)
8. Horn, B.K.P.: Height and gradient from shading. IJCV 5(1), 37–75 (1990)
9. Horn, B.K.P., Brooks, M.J.: Shape from Shading. MIT Press (1989)
10. Horn, B.K.P., Woodham, R.J., Silver, W.M.: Determining shape and reflectance using multiple images. Technical Report AI Memo 490. MIT (1978)
11. Kampel, M., Sablatnig, R.: 3D puzzling of archeological fragments. In: Skocaj, D. (ed.) Proc. of 9th Computer Vision Winter Workshop, pp. 31–40 (2004)
12. Kirkpatrick, D.G.: Optimal search in planar subdivisions. SIAM J. on Computing 12, 28–35 (1983)
13. Smith, L.N., Hansen, M.F., Atkinson, G.A., Smith, M.L.: 3D face reconstructions from photometric stereo using near infrared and visible light. Computer Vision and Image Understanding 114, 942–951 (2010)
14. Ng, Wu, Tang: Surface-from-gradients without discrete integrability enforcement: A Gaussian kernel approach. IEEE TPAMI 32 (November 2010)
15. Robles-Kelly, A., Hancock, E.R.: Surface height recovery from surface normals using manifold embedding. In: Proc. ICIP (October 2004)
16. Saracchini, Stolfi, Leitao, Atkinson, Smith: Multi-scale depth from slope with weights. In: Proceedings of the BMVC, pp. 40.1–40.12. BMVA Press (2010)
17. Smith, G.D.J., Bors, A.G.: Height estimation from vector fields of surface normals. In: Proc. IEEE DSP, pp. 1031–1034 (2002)
18. Smith, M.L., Smith, L.N.: Polished Stone Surface Inspection using Machine Vision, page 33. OSNET (2004)
19. Terzopoulos, D.: The computation of visible-surface representations. IEEE TPAMI 10(4), 417–438 (1988)
20. Terzopoulos, D.: Image analysis using multigrid relaxation methods. IEEE TPAMI PAMI 8(2), 129–139 (1986)
21. Wei, T., Klette, R.: Height from gradient using surface curvature and area constraints. In: Proc. 3rd ICVGIP (2002)
22. Woodham, R.J.: Photometric method for determining suface orientation from multiple images. Optical Engineering 19(1), 139–144 (1980)

Virtual Viewpoint Disparity Estimation and Convergence Check for Real-Time View Synthesis

In-Yong Shin and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Korea
{siy0808,hoyo}@gist.ac.kr

Abstract. In this paper, we propose a new method for real-time disparity estimation and intermediate view synthesis from stereoscopic images. Some 3D video systems employ both the left and right depth images for virtual view synthesis; however, we estimate only one disparity map at a virtual viewpoint. In addition, we utilize hierarchical belief propagation and convergence check methods to find the global solution rapidly. In order to use the virtual viewpoint disparity map for intermediate view synthesis, we build an occlusion map that describes the occlusion information in the virtual viewpoint region of the reference image. We have also implemented the total system using GPU programming to synthesize virtual viewpoint images in real time.

Keywords: Stereo matching, belief propagation, CUDA, DIBR, GPU programming, view interpolation.

1 Introduction

In recent years, various researches have been on a 3D video system as increasing interest in a 3D multimedia service. The 3D video system provides realistic multimedia services that offer 3D effects based on a binocular depth cue. It can be used in a wide range of multimedia applications such as immersive games, movies, presentations, video conferencing, 3D TVs and medical imaging. With the increasing demand of a 3D video display, MPEG has made an effort for a 3D audio-visual (3DAV) technology standardization [1]. The information of the 3D video display is characterized by a disparity map that consists of disparity vectors (DVs) for pixel pairs between the left and right images. As shown in Figure 1, virtual viewpoint images can be synthesized with respect to different virtual camera positions using the disparity map. Thus, disparity map estimation and virtual view synthesis are two most important parts in 3D video display.

Many disparity map estimation algorithms for stereo image pairs have been proposed in the past, and they can be classified into two types. One type emphasizes a low computational complexity for real time implementation. The block matching algorithm (BMA) provides a good example for this type. Due to the low complexity, a quality of the resulting disparity map is lower and the low quality disparity map affects a quality of synthesized virtual view. The other type attempts to get an accurate disparity map with a higher complexity. For example, global energy

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 121–131, 2011.

minimization algorithms are proposed in [2-4] for this purpose. Even though these methods can be used to synthesize high quality virtual viewpoint images, they demand a large amount of computation. So, their real time implementation is challenge.

Most virtual viewpoint image generation methods use two disparity maps (left and right viewpoints) or single disparity map at one of two reference viewpoints. First methods generate accurate synthesized image at a virtual viewpoint. However, it takes a long time to estimate two disparity maps. Second method needs half time for disparity estimation, but synthesis accuracy is lower than first one due to occlusion regions.

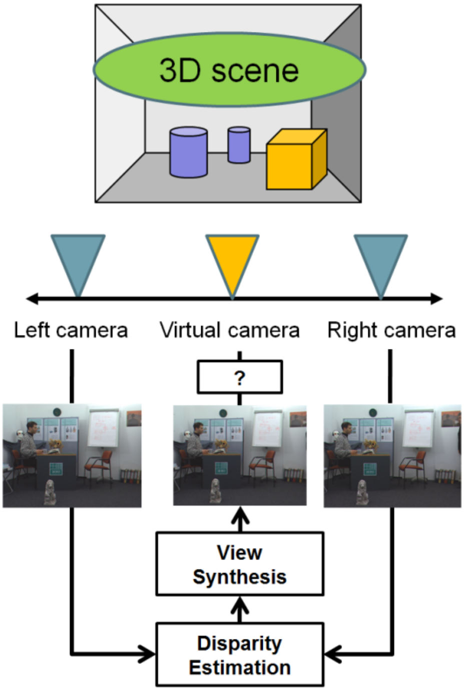


Fig. 1. Outline of view synthesis method

In this paper, we propose a real time virtual viewpoint synthesis method. In order to synthesize virtual viewpoint images in real time, we estimate disparity maps at the virtual viewpoint. Also we find convergence regions of the disparity map in the hierarchical belief propagation process. We cancel message updates at convergence regions to remove residual calculation by using a convergence map. After the disparity estimation process, we decide the occlusion map of the virtual viewpoint to select regions which can be back-projected. We synthesize the virtual viewpoint image using the virtual viewpoint disparity map and the occlusion map. Additionally, we implement the proposed method in real time using parallel programming called

CUDA. CUDA is the general purpose computing engine in NVIDIA GPUs that is accessible to software developers through industry standard programming languages.

This paper organized as follows. In Section 2, related work about view interpolation is explained. In Section 3, our proposed method is explained. In Section 4, the experimental results are given. The conclusion is presented in Section 5.

2 Related Work

There are many researches related to virtual viewpoint image synthesis techniques. Generally, left and right disparity maps are used for view synthesis [5]. As shown in Figure 2, this method estimates two left and right disparity maps and warp virtual images respectively. Then, two virtual images are summed by weighting function. Although it has heavy complexity due to two disparity estimation parts, it generates virtual images which are respectable quality.

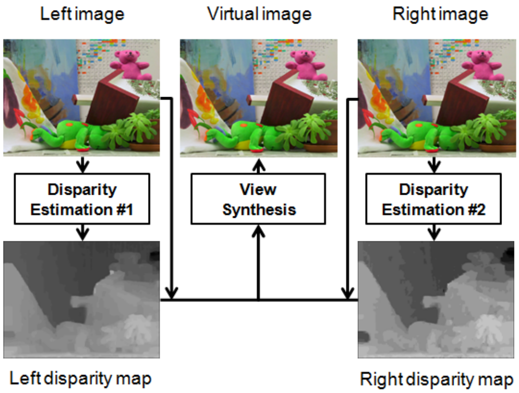


Fig. 2. Conventional view synthesis method

Also, single disparity map estimation process can be used to synthesize a virtual viewpoint image. For instance, the single disparity map at left or right viewpoint can be used to generate virtual viewpoint image [6]. Omitting a disparity map estimation part of the other viewpoint leads it to fast execution. Qualities of view synthesis outputs are, however, lower than the first method due to occlusion regions which should only refer pixel information from the other viewpoint.

In the global disparity estimation methods, the belief propagation algorithm is frequently used [7]. Although it produces an accurate disparity map, it is too slow to be practical. So, the hierarchical belief propagation algorithm is proposed [8][9]. It runs much faster than the previous algorithms while maintaining comparable accuracy. The main difference between the HBP and the standard BP algorithm is that the HBP algorithm works in a coarse-to-fine manner. In other words, the HBP algorithm estimates the disparity map with a smallest resolution, then it estimates higher resolution disparity maps with a previously estimated disparity map. The basic steps are: (a) initialize the messages at the coarsest level to all zeros, (b) apply the BP algorithm at the coarsest level to iteratively refine the messages, (c) use refined messages from the coarser level to initialize the messages for the next level. Specifically, if X is a pixel at a coarser level, and its corresponding pixels at the finer level are $X'_{\mathrm{i,\,i}} \in [1, 4]$, as shown in Figure 3.

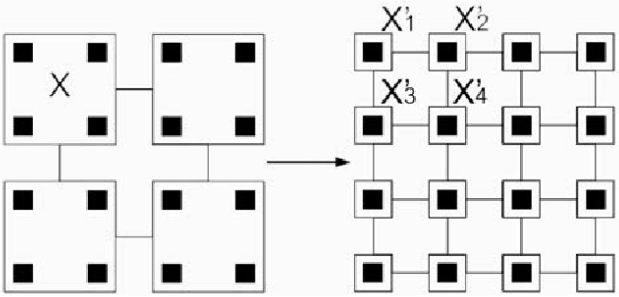


Fig. 3. Two levels in the coarse-to-fine method

Two main parameters S and T define the behavior of the HBP algorithm, S is the number of levels and T is the number of iterations at each level. Generally, we estimate disparity maps with five levels and ten iterations (S=5, T=10). Actually, we only compute beliefs (disparity map) at level 0 in the HBP algorithm.

3 Virtual Disparity Estimation and Convergence Check

In this section, we describe the proposed method for the real time virtual viewpoint image generation using the virtual viewpoint disparity estimation method and the convergence check method of the HBP algorithm. As shown in Figure 4, our method contains following steps.

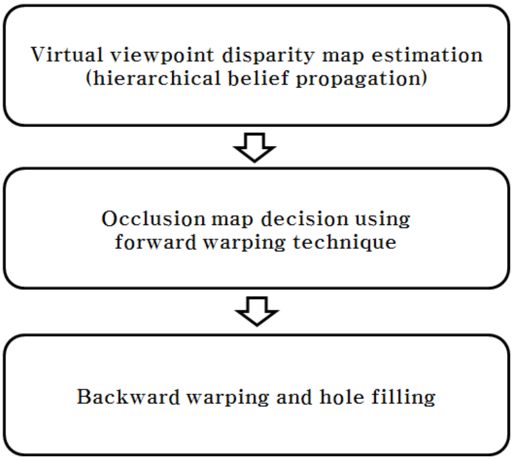

Virtual viewpoint disparity map estimation
(hierarchical belief propagation)
Occlusion map decision using
forward warping technique
Backward warping and hole filling

Fig. 4. View synthesis using virtual disparity estimation

3.1 Virtual Viewpoint Disparity Estimation

Global stereo matching methods find corresponding points using iterative energy minimization algorithms. An energy function E considers photo-consistency (a corresponding pixel should have the same intensity value) and piecewise smoothness (neighboring pixels are likely to have the similar disparity value).

$$E(x,y,d) = E_{data}(x,y,d) + E_{smooth}(x,y,d) \tag{1}$$

As shown in Figure 5, we directly estimate the disparity map at the virtual viewpoint. For this case, we calculate data cost by using

$$\sum_{x,y}\left|I_R\left(x + d_{V\_L}(x,y),y\right) - I_R\left(x - d_{V\_R}(x,y),y\right)\right| \tag{2}$$

where $d_{V\_L/R}$ and $I_{L/R}$ are virtual viewpoint disparity maps and input images. Relationship between disparity values of d_{V_R} and $d_{\mathrm{V\_L}}$ is

$$d_{V\_R}(x,y) = Alpha \times d_{V\_L}(x,y) \tag{3}$$

where *Alpha* is a relative distance from the virtual viewpoint to the right viewpoint when a distance between the left viewpoint and the virtual viewpoint is one.

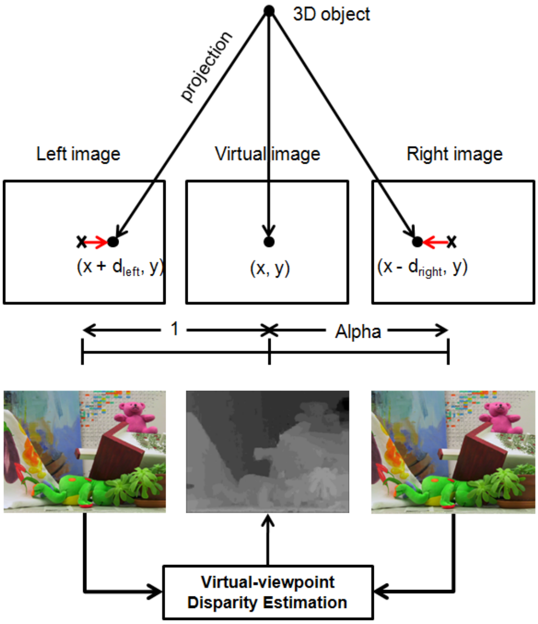


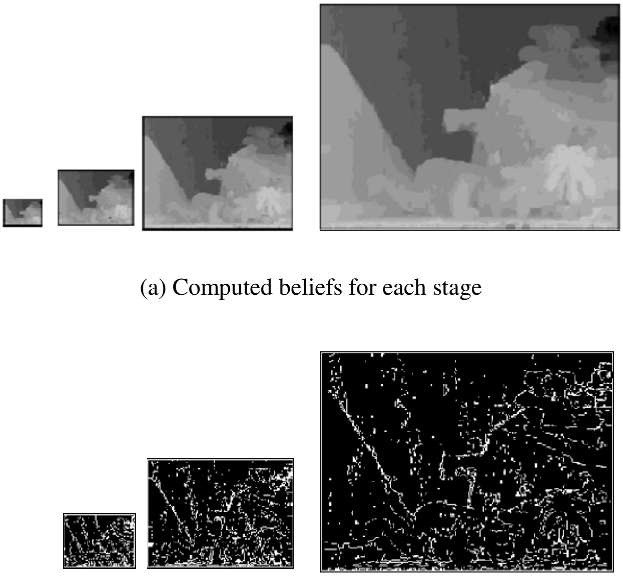
Fig. 5. Virtual viewpoint disparity estimation

The hierarchical belief propagation algorithm is used to minimize the energy function. It passes messages called belief around in four adjacency image grids. Message updates are in iterations. At one iteration step, each pixel of the adjacency graph computes its message based on the message to all the adjacent pixels.

3.2 Acceleration of HBP Using Convergence Check

We can find convergence regions of message in the HBP. After each level shifting, we can find it by comparing midterm result of the HBP at each level. Then, we can stop message updates at convergence regions, if differential values of the message are lower than a predetermined threshold value. However, it takes non-slight additional time to check convergence regions due to many message paths (four directions, up, down, left, and right, message passing paths).

In order to obtain the convergence map efficiently, we compute beliefs (disparity map) for each stage and we check whether it stationary during a stage transition. Actually, beliefs are calculated only at a zero stage in the standard HBP algorithm. After we build the convergence map, we subtract converged nodes from the message update process. Figure 6(a) shows computed beliefs for each stage which is ‘Teddy’ which is given by the Middlebury web site. Figure 6(b) are convergence maps. In the convergence map, black and white pixels mean converged region and non-convergence region.



(a) Computed beliefs for each stage

(b) Binary convergence map (block: converged, white: non-converged)

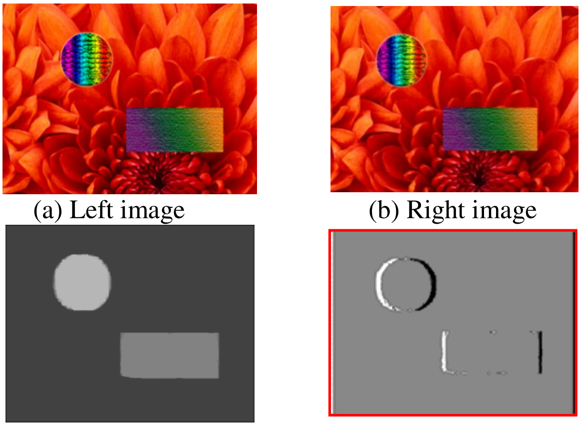
Fig. 6. Convergence check of HBP

3.3 Occlusion Decision and Backward Warping

If we have the virtual viewpoint disparity map, we can make the virtual viewpoint image using a backward warping process. Before we warp reference images to virtual viewpoint image plane, we have to consider possible errors due to occlusion regions. In order to avoid a occlusion problem of backward warping, we need to check occlusion regions of the virtual viewpoint from left and right viewpoints. So, we decide an occlusion map which is composed with four labels.

$$O_v(x,y) = \begin{cases} \text{A, Occluded from } I_L \\ \text{B, No occlusion} \\ \text{C, Occluded from } I_R \\ \text{D, Occluded from } I_L \text{and } I_R \end{cases} \quad (4)$$

The occlusion map can be labeled by forward warping of a virtual viewpoint disparity map to left and right viewpoints. Label *A* can be selected when pixel information of the virtual viewpoint is only occluded from the left viewpoint. Label *B* can be selected when pixel information of the virtual viewpoint is not occluded from any viewpoints. Label *C* can be selected when pixel information of the virtual viewpoint is occluded only from the right viewpoint. In the case of label *D*, virtual viewpoint pixel information is occluded both viewpoints. Figure 7 shows an input stereo image pair, the virtual disparity map, and the occlusion map which has four labels (255-A, 128-B, 0-C, 1-D). Artificial images are used to see occluded regions clearly.


(a) Left image

(b) Right image

(c) Disparity map

(d) Occlusion map

Fig. 7. Occlusion map decision

3.4 Hole Filling

Although warping process fills up proper pixel values from the reference images, there are still unknown hole regions which cannot find a same fetch from the reference images due to a occlusion problem. Thus, we have to fill the hole with the most plausible value by using surrounding pixel information. Most of the presented hole filling methods use image interpolation or in-painting algorithm. In order to get best quality hole filled images, neighboring background pixel values and their geometric information should be used. The reason why we use generally background region information is that background pixels rather than the foreground ones as the disoccluded area is more reasonable by definition of the disocclusion [10,11]. Thus, we fill up hole regions with neighboring pixel values which have background disparities.

3.5 GPU Implementation

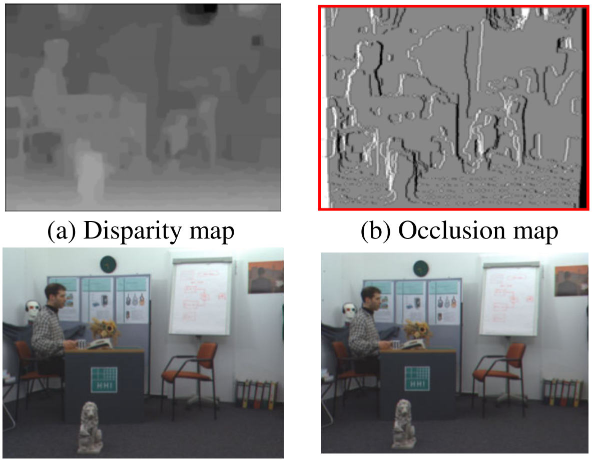
For the real time implementation, we use the GPU parallel programming which executed on the GPU. The architecture of CPU and GPU are very different. Although GPU has a small number of instruction control unit, it has a lot of cores capable of calculating floating points operation. Thus, GPU has a Single Instruction Multiple Threads (SIMT) structure [12]. So, image processing algorithm is very suitable for GPU programming due to that all of image pixels may have same operation. There is an important condition of the SIMT parallel processing. It is a data independency between all data executed simultaneously. We implement whole process with the parallel GPU programming while maintaining a data independency.

4 Experimental Results

In order to evaluate performance of proposed algorithm, we have implemented three methods (method A, method B, and proposed method) on CPU and additionally applied GPU parallel programming to proposed method. Because fast processing time and acceptable visual quality are key points of our algorithm, we measured processing time and visual quality by calculating PSNR value between original and output images. Furthermore, we check these measurements with other two methods. Method A and B are conventional methods. Method A interpolates the virtual viewpoint image using left and right disparity maps. Method B uses only a left disparity map. For the experiment, we performed tests on several rectified stereo images which listed in Table 1. Test images are obtained from Middlebury stereo website and MVD test materials. Test stereo set includes not only stereo images, but also intermediate viewpoint images to verify synthesis quality by comparing original images.

Table 1. Specification of the test stereo image set

Sequence	Teddy	Poster	Cones	News papers	Book arrivals
Size	640x480	480x416	480x416	640x480	640x480
max disparity	30	20	20	50	50



(a) Disparity map

(b) Occlusion map

(c) Original image

(d) Virtual-view image

Fig. 8. Input and output images

Figure 8 shows input and output images of the proposed method. In order to investigate synthesis quality, we calculate PSNR values with synthesized images and original intermediate viewpoint images. Figure 9 and 10 shows performance comparison of the three methods. Results prove that our proposed method is accurate and faster than others. Moreover, implementation of GPU parallel programming carries additional speed up as shown in Figure 11. We implement same algorithm on the CPU and GPU. However GPU execution speed is better than CPU because GPU execute multiple threads at the same time. As a result, GPU accelerates execution speed up to 30 times by comparing CPU execution time.

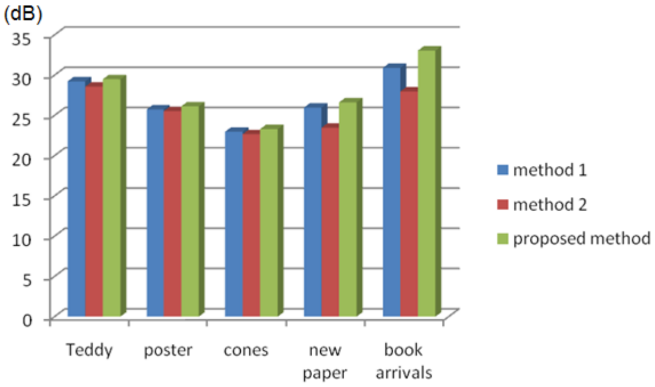


Fig. 9. PSNR comparisons of three methods

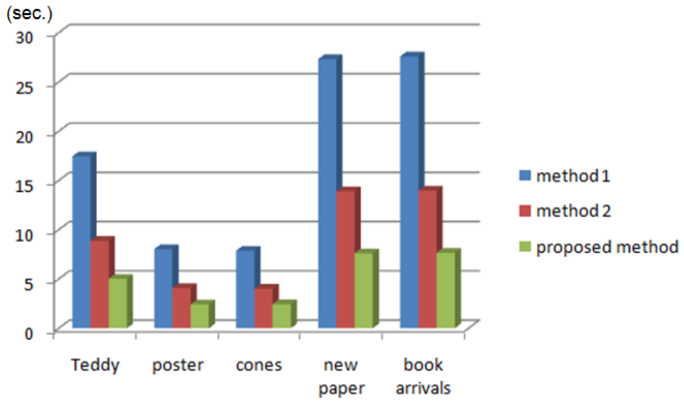


Fig. 10. Execution time of three methods

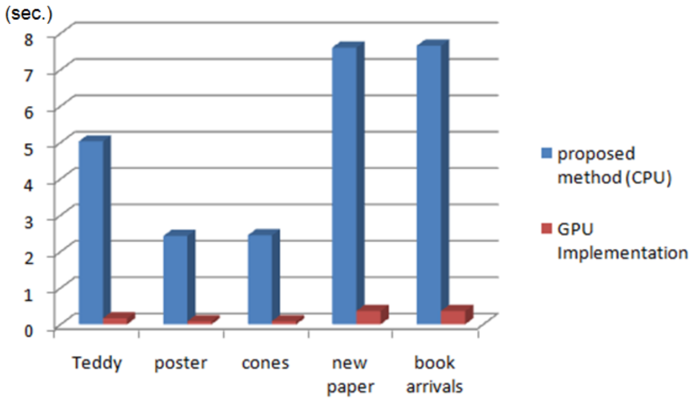


Fig. 11. Execution time in CPU and GPU

5 Conclusions

In this paper, we present the real time view interpolation method. In order to make it more rapidly, we apply the virtual viewpoint disparity estimation method and GPU parallel programming. Previous methods estimate some duplicated and unnecessary disparity values for the certain viewpoint. Thus, our proposed method reduces complexity and makes accurate synthesized images by eliminating surplus calculation. We designed the data cost function for the virtual viewpoint disparity map. The hierarchical belief propagation algorithm is used to minimize the energy function. In the view synthesis part, we warp pixels from reference images to the virtual viewpoint using the virtual viewpoint disparity map. In order to check a synthesized image quality, we calculate PSNR values by comparing original images and synthesized images. Our results are generally 0.3dB higher than previous method. For the real time implementation, we utilize the high speed GPU parallel programming called CUDA. As a result, we can synthesize the virtual viewpoint image at a rate of 30 frames per second at most.

Acknowledgments. This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-(C1090-1111-0003)).

References

1. ISO/IEC JTC1/SC29/WG11 N6909: Survey of algorithms used for multi-view video coding, MVC (2005)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 1222–1239 (2000)
3. Kolmogorov, V., Zabih, R.: Multi-Camera Scene Reconstruction via Graph Cuts. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 82–96. Springer, Heidelberg (2002)
4. Sun, J., Zheng, N., Shum, H.: Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 787–800 (2003)
5. ISO/IEC JTC1/SC29/WG11 M1537: Contribution for 3D Video Test Material of Outdoor Scene (2008)
6. Oh, J., Ma, S., Kuo, C.: Disparity estimation and virtual view synthesis from stereo video. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, LA, USA, pp. 993–996 (2007)
7. Tappen, M., Freeman, W.: Comparison of Graph Cuts with Belief Propagation for Stereo. In: Proc. IEEE Int'l Conf. Computer Vision, vol. 1, pp. 508–515 (2003)
8. Felzenszwalb, P., Huttenlocher, D.: Efficient Belief Propagation for Early Vision. In: CVPR, vol. 1, pp. 261–268 (2004)
9. Yang, Q., Wang, L., Yang, R., Stewenius, H., Nister, D.: Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 492–504 (2009)
10. Oh, K., Yea, S., Ho, Y.: Hole Filling Method using Depth Based In-painting for View Synthesis in Free Viewpoint Television and 3-D Video. In: Picture Coding Symposium, pp. 39 (1-4) (2009)
11. Oliveira, M., Bowen, B., McKenna, R., Chang, Y.: Fast Digital Image Inpainting. In: Proceedings of the International Conference on Visualization, Imaging and Image Processing, Marbella, Spain (2001)
12. NVIDIA Corporation, CUDA 3.2 Programming Guide (2010), `http://www.nvidia.com/cuda_develop.html`

Spatial Feature Interdependence Matrix (SFIM): A Robust Descriptor for Face Recognition

Anbang Yao[1] and Shan Yu[2]

[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Science, Beijing, 100090, China
[2] National Institute for Research in Computer Science and Control, France
abyao@nlpr.ia.ac.cn, shanyu329@yahoo.com

Abstract. In this paper, a new face descriptor called spatial feature interdependence matrix (SFIM) is proposed for addressing representation of human faces under variations of illumination and facial expression. Unlike traditional face descriptors which usually use a hierarchically organized or a sequentially concatenated structure to describe the spatial arrangement of features in different facial regions, SFIM is focused on exploring inherent spatial feature interdependences among separated facial regions in a face image. We compute the feature interdependence strength between each pair of facial regions as the Chi square distance between two corresponding histogram based feature vectors. Once face images are represented as SFIMs, we then employ spectral regression discriminant analysis (SRDA) to achieve face recognition under a nearest neighbor search framework. Extensive experimental results on two well-known face databases demonstrate that the proposed method has superior performance in comparison with related approaches.

Keywords: Face recognition, spatial feature interdependence matrix, spectral regression discriminant analysis, object representation.

1 Introduction

Urged by the fact that human face is one of the most potential physiological biometrics [1] for many applications such as public security, surveillance, human computer interaction (HCI) and multimedia, automatic face recognition has been an active research area in computer vision community for over three decades [2], [3], [4]. From the perspective of practical application, a desirable 2-D image based face recognition system should be the one that is able to identify or verify a human face under variations of facial expression, illumination, pose and occlusion in an accurate and efficient manner. To this end, a myriad of approaches have been proposed so far. However, face recognition under above challenges is still far from being effectively resolved [3], [5], [6]. One critical reason that prevents this from happening is the lack of reliable and generic methods to represent face instead of image data itself. The fact of the matter is that the information contained in source image is usually highly redundant or non-discriminative for face recognition regardless of feature extraction.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 132–143, 2011.

A lot of seminal research works have already been done on the issue of how to build up an effective and compact face representation from 2-D image feature space. The current state of the art approaches differ vastly in terms of principles and techniques applied. Following a high-level categorization guideline suggested from the psychological studies on how human recognize objects, the available face representation approaches can be classified into two categories—on the basis of holistic information and on the basis of local features [3], [7], [8], [9].

The category of holistic face representation approaches is characterized by a family of subspace methods originated from the Eigenface approach [10], which employs principal component analysis (PCA) to project high dimensional face feature vectors onto a significantly low dimensional feature space. By probing the projection directions that maximize the total scatter of all labeled face data of the same person, the underlying Euclidean space structure is discovered. Different from PCA, linear discriminant analysis (LDA) [11] pays particular attention to the discrimination between face classes in a linear separable space without the prerequisite of orthogonal bases. Although PCA and LDA are the two most popular techniques for face recognition, they cannot recover the non-linear structure of data set. To address this problem, the Laplacianface method [12] employs locality preserving projections (LPP) to find an embedding that preserves local information, and obtains a face subspace that best detects the essential face manifold structure. Similar to the approaches of [10], [11] and [12], other popular holistic face representation approaches including kernel based methods [13], [14], independent component analysis (ICA) [15], 2-dimensional principal component analysis (2-D PCA) [16] and so on, are also designed to address the problem of linear/nonlinear dimensionality reduction. Apart from these traditional holistic representation approaches [10], [11], [12], [13], [14], [15], [16], a probe face image can also be expressed as various linear combinations of gallery set [17], [18], [19]. In summary, all above mentioned holistic face representation methods directly use typical intensity images of human face as the inputs. That is, they mainly focus on seeking a linear/non-linear subspace or a linear combination that can best represent the structure of face data in favor of the class with minimal reconstructive error, while pay less attention to the fact that gray-scale images are generally not discriminative enough owing to the effect of factors such as redundancy, rotation, noise, lighting and albedo.

The category of local face representation approaches has recently also gained attention due to its robust capability to handle difficulties such as rotation and lighting. In general, this category approach benefiting from local information assumes that individual features extracted from prominent facial regions (e.g., eyes, nose, chin and mouth) are more vital to face recognition than the identification of holistic information. One of the pioneering works is elastic bunch graph matching (EBGM) [20], which describes faces using Gabor filter responses in 25 facial landmarks and uses a graph structure to represent the spatial locations of these landmarks. A modified EBGM algorithm is presented in [21] where the Gabor features in all facial landmarks are replaced by the histograms of oriented gradients (HOG) [22]. Instead of using a graph structure to represent face, the authors of [23] consider face images as a composition of micro patterns over small regions and take spatially weighted local binary pattern (LBP) histograms as the descriptors to represent face images. Later on, local Gabor binary pattern histogram sequence (LGBPHS) [24] and histogram of Gabor phase patterns (HGPP) [25] are proposed for robust face recognition by the same research group. LGBPHS is intended to use the magnitude parts of Gabor filter and LBP operator simultaneously, while HGPP is mainly designed to jointly encode local and global Gabor phase patterns. In the approaches of [23], [24] and [25], a face image is divided into different regions, from which respective histograms are independently extracted and further concatenated into an extended histogram vector to represent the target face. It is worth noting that these approaches put more emphasis on the spatial arrangement of features in facial regions yet pay less attention to the spatial feature interdependences among different facial regions.

In this paper, we explore inherent spatial feature interdependence between any two different facial regions in a face image. By encoding all pair-wise spatial feature interdependence strengths over separated facial regions inside a face image, we propose a new face descriptor called spatial feature interdependence matrix (SFIM). In contrast to previous face descriptors [23], [24], [25], SFIM explicitly depicts an inherent spatial feature interdependence network among the facial parts of a face image. Furthermore, it provides a bridge to the association of sophisticated learning approaches [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] and discriminative local features [22], [23], [24], [25]. In our approach, the feature interdependence between two different facial regions is estimated as the Chi square distance between a corresponding histogram vector pair. Once face images are represented as SFIMs, spectral regression discriminant analysis (SRDA) [26] is further employed to achieve face recognition under a nearest neighbor search framework. Comparative experiments are confined to frontal human face recognition, and the variations of illumination and facial expression are mainly addressed. Extensive experimental results on the extended Yale B [27] and the cropped AR [28] face databases validate the efficacy of the proposed approach.

The rest of this paper is organized as follows. Section 2 presents a detailed description of our SFIM based face descriptor. Section 3 describes the face recognition algorithm. Section 4 presents the comparative experiments on two publicly available face databases. Section 5 summarizes the paper and makes an outlook of possible future extensions.

2 Spatial Feature Interdependence Matrix

In this section, we will define the concept of SFIM, describe SFIM based descriptor for face object and show the properties of SFIM.

2.1 Definition of SFIM

The idea of SFIM is inspired by the following two facts. First, a standard human face is a whole unit consisting of different facial part structures. A successful application example of this fact is the well-known facial action coding system (FACS) [29], in which 46 different action units are defined to account for changes in facial expression. Second, as for human visual system, face recognition is not only dependent on visual information extracted from prominent facial parts (e.g., eyebrows, eyes, nose, mouth and chin) but also assisted by latent information contained in the non-prominent facial parts [2], [3], [5], [18]. The aim of our proposed SFIM is not focused on precise segmentation of facial parts but on exploring spatial feature interdependence between any two different facial regions in a face image. We believe that the spatial feature interdependence between each pair of facial regions is a potential cue to identify face images of a certain person, and a careful handling of spatial feature interdependences may provide a completely new face representation approach. In accordance with above description, SFIM is defined as follows.

Let I represent a face image containing M separated regions $\{R_1, R_2, \cdots, R_M\}$, the SFIM of I is defined as a square symmetric matrix of size $M \times M$.

$$A = [a_{ij}]_{M \times M} = \begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1M} \\ a_{21} & 0 & a_{23} & \cdots & a_{2M} \\ a_{31} & a_{32} & 0 & \cdots & a_{3M} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{M1} & a_{M2} & a_{M3} & \cdots & 0 \end{bmatrix}, \tag{1}$$

where a_{ij} is the feature interdependence between facial regions R_i and R_j, $0 \le a_{ij} \le 1$, $1 \le i, j \le M$. The diagonal corresponds to the feature interdependence of a facial region itself, and it is composed of zeros.

2.2 SFIM Based Face Descriptor

From equation (1), it is clear the exact patterns of SFIM directly rely on the features and measures used to compute a_{ij}. For easy implementation, in our approach, face images are divided into spatially non-overlapped rectangular facial regions of same size. As the facial regions $\{R_1, R_2, \cdots, R_M\}$ have been determined, a feature vector is computed in each facial region independently. Various features are available in the literature for describing an image region of interest. We employ widely used histogram features to describe each rectangular facial region mainly due to two reasons. First, as the quantized and compact distributions of particular contents (e.g., intensity, gradient, phase, texture and high order filter response) in an image region of interest, histogram features have already been shown to be robust to noise, local image transformation, partial occlusion, etc. Second, histogram features can be calculated in a highly efficient way.

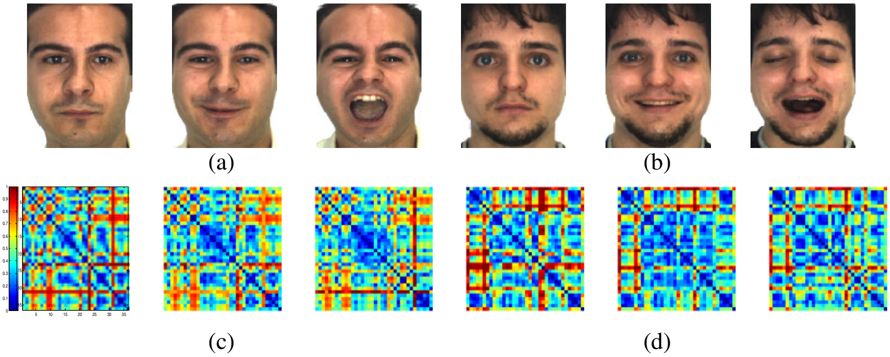
Let h_i and h_j be the normalized N-bin histograms of facial regions R_i and R_j, respectively, where $\sum_{n=1}^{N} h_i(n) = 1$ and $\sum_{n=1}^{N} h_j(n) = 1$. The corresponding feature interdependence between facial regions R_i and R_j is computed as

$$a_{ij} = \sum_{n=1}^{N} \frac{\left(h_i(n) - \bar{h}(n)\right)^2}{\bar{h}(n)}, \tag{2}$$

where

$$\bar{h}(n) = \frac{h_i(n) + h_j(n)}{2}. \tag{3}$$

In the literature, above defined interdependence is known as Chi square distance between a histogram feature vector pair. It measures how unlikely histogram distribution h_j is drawn from the population represented by histogram distribution h_i.



(a) (b)

(c) (d)

Fig. 1. The discriminative capability of SFIM on different face objects

Fig. 1 shows some examples to demonstrate the power of SFIM for representing faces under different facial expressions. The source images of two persons in three different facial expressions are displayed in Fig. 1(a) and Fig. 1(b), respectively. The corresponding SFIMs are shown in Fig. 1(c) and Fig. 1(d), respectively. The source images are taken from the cropped AR face database [28]. To compute corresponding SFIMs, the converted gray-scale images are divided into 6×6 spatially non-overlapped rectangular facial regions, and 32-bin intensity histogram is used to describe each facial region. Note that the patterns of the resulting SFIMs of each person are relatively similar across different facial expressions, but are different across two persons. These results preliminarily show that the proposed SFIM can effectively handle the difficulties resulted from facial expression changes. The detailed experiments described in section 4 will further show the effectiveness of our SFIM based face descriptor, and the choice of dimensionality of SFIM is also clarified in section 4.

2.3 Properties of SFIM

The above section presents the definition of SFIM and describes the approach to represent a face image as an SFIM. In this section, we will shed more light on the properties of our SFIM based face descriptor.

First, considering each facial region as a node, the most characteristic property of SFIM is that it depicts a feature interdependence network among the separated facial regions in a face image. That means all pair-wise spatial feature inconsistencies are explicitly encoded via SFIM. Consequently, different from traditional face descriptors which usually employ a hierarchically organized [20], [21] or a sequentially concatenated structure [23], [24], [25] to describe the spatial arrangement of features in different facial regions, SFIM is designed to make a careful handling of inherent spatial feature interdependence between any two different facial regions in a face image. Second, SFIM has a form of square symmetric matrix and its entries are computed from histogram features in different facial region pairs, which make the resulting SFIMs of face images can be easily used as the inputs of any sophisticated learning approaches [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] to achieve face recognition goal. In this paper, although we employ histogram features and Chi square distance to calculate the entries of SFIM, other features (e.g., covariance matrices [30] and Gabor based region covariance matrices [31]) and measures (e.g., Kullback-Leibler divergence and earth mover's distance) can also be used to compute SFIM. Therefore, the proposed SFIM provides a bridge to the association of sophisticated learning approaches and discriminative local features. Third, according to the description provided in section 2.2, it is clear that the computation of square symmetric SFIM is intrinsically efficient. Finally, the notations of video self-similarity matrices [32], [33], [34] used for describing periodic motion in temporal space are most closely related to our proposed SFIM in form. However, the main focus of SFIM is on the use of a structured layout of spatial feature interdependences over all pairs of facial regions for face representation. To our knowledge, this problem is not clearly addressed in available face representation methods [35], [36], [37], [38].

3 SFIM Based Face Recognition

Once every face image is represented as an SFIM with its entries computed as Chi square distances between histogram vectors extracted from any pair of non-overlapped rectangular facial regions, face recognition can be achieved in various ways. One possible way is to use a matrix measure (e.g., matrix cosine similarity and Frobenius norm) to directly compare the dissimilarity between a probe SFIM and every gallery SFIM. Another possible way is to use the SFIMs as the inputs of a traditional subspace method [10], [11], [12], [13], [15], [26] or a linear optimization approach [17], [18], [19]. Considering that SRDA [26] casts discriminant analysis into a regression framework that facilitates efficient computation, we use it as a test case to evaluate the performance of SFIM based face descriptor under a nearest neighbour search framework. Recall that SFIM is a square symmetric matrix whose diagonal entries are zeros, the lower/upper triangular entries of SFIM are discarded.

Let us consider a gallery set of N normalized face images belonging to C classes, and assume that n_k represents the number of training images of the kth face class ($\sum_{k=1}^{C} n_k = N$). The face recognition algorithm is described as follows.

1. Compute the SFIM of every gallery face image according to section 2.
2. Rearrange the upper triangular entries (excluding diagonal) of each SFIM into a vector. Let $x_1, x_2, \cdots, x_N \in \Re^{(M^2-M)/2}$ be the resulting vector set, where $M \times M$ is the dimensionality of SFIM.
3. Let $y_k = \left[\underbrace{0,\cdots,0}_{\sum_{i=1}^{k-1} n_i},\underbrace{1,\cdots,1}_{n_k},\underbrace{0,\cdots,0}_{\sum_{i=k+1}^{C} n_i}\right]^T, k = 1,\cdots,C$ and $y_0 = [1,1,\cdots 1]_{1\times N}^T$, find $C-1$ basis vectors $\{v_k\}$ by solving

$$v_k = \arg\min_{v}\left(\sum_{i=1}^{N}\left(v^T x_i - y_i^k\right)^2 + \alpha\|v\|^2\right), \tag{4}$$

where $v_k \in \Re^{(M^2-M)/2}$, y_i^k is the ith element of Gram-Schmidt orthogonalized y_k.

4. Embed each training x_i into the $C-1$ dimensional subspace by

$$z_i = V^T\begin{bmatrix} x_i \\ 1 \end{bmatrix}, \quad V = \begin{bmatrix} v_1, v_2, \cdots v_{C-1} \\ 1, \ 1, \cdots 1 \end{bmatrix}. \tag{5}$$

5. Given a probe face image, compute corresponding SFIM according to step 1 and rearrange it into a vector x according to step 2.
6. Get the embedding z of x by equation (5).
7. Compute the Euclidean distance between z and each z_i, and take the class of the gallery face image with the minimum distance to label probe face image.

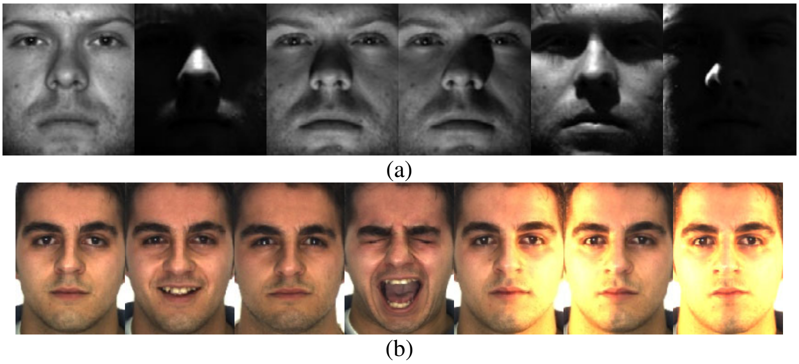
4 Experiments and Results

In this section, extensive experiments on two challenging face databases (i.e., the extended Yale B [27] and the cropped AR [28] face databases) are carried out to demonstrate the efficacy of the proposed SFIM. In the experiments, two kinds of histogram feature (i.e., 32-bin intensity histogram and histogram of LBP) are employed to calculate SFIM. A comprehensive comparison of the performance of our approach and the published results on each test database is presented. We want to point out here that the reported results grouped in [18] (including Eigenface [10], Fisherface [11], Laplacianface [12], SVM+Laplacianface [14] and sparse representation [18]) are shown for a large range of feature dimensions. For the sake of fair comparison, we just pick the best results. The results of spatially weighted LBP

[23], LGBPHS [24] and the histogram of monogenic binary pattern (HMBP) [39] on the cropped AR face database are collected from [39]. Considering that we embed SFIM based descriptor in SRDA algorithm to achieved face recognition, the results of standard SRDA [26] are also presented. As for standard SRDA, the results are computed for an exact same range of feature dimensions to that of [18], and only the best results are shown for comparison (see the supplementary file for details).

4.1 Experiments on the Extended Yale B Face Database

The extended Yale B face database contains 2414 images of 38 subjects showing 1 frontal pose under 64 laboratory-controlled illumination conditions. Some examples are shown in Fig. 2(a). The cropped and normalized gray-scale image has a resolution of 192×168 pixels. For each subject, half of the images are selected for training while the other half are adopted as testing dataset. All these parameters are same to those of the published reference results grouped in [18].



(a)

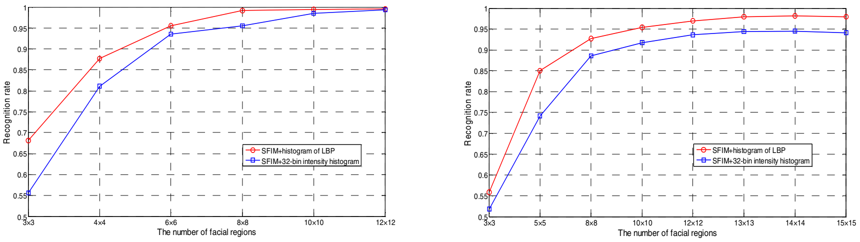
(b)

Fig. 2. Examples of representative subjects from: (a) the extended Yale B face database; (b) the cropped AR face database

Table 1. Comparison of the top recognition accuracy of different approaches

Methods	Recognition rate on the extended Yale B face database (%)	Recognition rate on the cropped AR face database (%)
Eigenface [10]	88.40	80.50
Fisherface [11]	87.60	86.80
Laplacianface [12]	90.70	89.70
SVM+Laplacianface [14]	97.70	95.70
Sparse representation [18]	98.26	94.99
Spatially weighted LBP [23]	N/A	97.71
LGBPHS [24]	N/A	97.29
HMBP [39]	N/A	98.57
Standard SRDA [26]	94.38	78.71
SFIM+32-bin intensity histogram	99.34	94.52
SFIM+histogram of LBP	99.59	98.71

The results on this database are shown in Table 1. From Table 1, it can be seen that the Fisherface method is worse than the others. This is partially due to the fact that the maximal number of valid Fisherfaces is one less than the number of face classes [11], [18]. Although the approaches of [10], [12] outperform the Fisherface method, their recognition rates are relatively low. The algorithm presented in [14] achieves a high recognition accuracy of 97.7%. More high result is reported for the sparse representation approach [18]. Taking the SFIMs computed from 32-bin intensity histogram and histogram of LBP as face descriptors, our approach yields recognition rates of 99.34% and 99.59%, respectively. These results are 1.08 and 1.33 percent better than the best result of comparison approaches, respectively. Furthermore, by utilizing intensity histogram based feature interdependence between each facial region pair in a face image, our approach outperforms standard SRDA (whose inputs are normalized intensity images) by 4.96 percent in accuracy. These results fairly verify our assumption that the spatial feature interdependence between each pair of facial regions is a potential cue to identify face images in difficult scenarios. Therefore, by properly encoding all pair-wise spatial histogram based feature interdependences over different facial regions inside a class-specific face image, SFIM can handle the difficulty resulted from illumination changes in a more effective way. The choice of dimensionality of SFIM on the extended Yale B face database is shown in Fig. 3(a). With respect to the SFIM based face descriptors computed from histogram of LBP, it can be seen that the recognition rate of our approach becomes constant (>99%) in the face partition range of 8×8~12×12 rectangular regions. As for the other case, stable results (>98.5%) are achieved in the face partition range of 10×10~12×12 rectangular regions.



(a) (b)

Fig. 3. The choice of the dimensionality of SFIM on: (a) the extended Yale B face database; (b) the cropped AR face database

4.2 Experiments on the Cropped AR Face Database

The cropped AR face database consists of 2600 color images corresponding to 100 subjects (50 men and 50 women). Different facial expressions (neutral, smile, anger and surprise), illumination conditions (left light on, right light on and all lights on) and occlusions (glass, scarf, etc.) are included. Each subject participates in two sessions separated by two weeks. The same images are taken in both sessions. For each subject, 14 images with illumination and expression variations are considered in the experiments. The 7 images from session 1 are selected for training, and the other

7 images from session 2 are chosen for testing. The cropped 165×120 color images (as shown for 7 example images in Fig. 2(b)) are converted to gray-scale images. Same to section 4.1, all above parameters are identical to those of the published results grouped in [18], [39].

The results on this database are shown in Table 1. Similar to the results on the extended Yale B face database, the approaches of Eigenface, Fisherface and Laplacianface exhibit relatively low recognition rates, while the approaches of [14] and [18] demonstrate better performance. More high results are reported for the approaches of [23], [24], [39]. As for the SFIM based face descriptors computed from histogram of LBP, our approach yields the best recognition rate of 98.71%. With regard to the SFIM based face descriptors computed from 32-bin intensity histogram, the recognition rate of our approach is 94.52%. Compared with the results of standard SRDA, the results of our approach are 20 and 15.81 percent better, respectively. These results fairly show that typical intensity images of human face are generally not discriminative enough for face recognition in complex scenarios. Since the inherent spatial feature inconsistencies between any two different facial regions of a class-specific face image are encoded via SFIM, our approach demonstrates better capability to deal with the variations of facial expression and illumination. The choice of dimensionality of SFIM on the cropped AR face database is shown in Fig. 3(b). Note that the recognition rate of our approach becomes stable in the face partition range of 13×13~15×15 rectangular regions (>98% and >94% corresponding to two types of SFIM, respectively).

5 Conclusions

This paper presents a new face descriptor called SFIM. SFIM takes advantage of the form of a square symmetric matrix to explicitly encode inherent spatial feature interdependence between any two separated facial regions in a face image. We first use SFIMs computed from histogram features and Chi square distance as the descriptors to represent face images. Subsequently, SRDA is used as a test case to achieve efficient face recognition under a nearest neighbor search framework. Extensive experiment results on two well-known benchmark face databases verify the efficacy of the proposed SFIM.

Our further work will be extended in two directions. First, how to incorporate more class-specific information into SFIM is an open issue. Second, SFIM can be generalized for representing other objects needed in applications such as object detection, multi-class object classification and content based image retrieval (CBIR).

References

1. Jain, A.K., Ross, A., Parbhakar, S.: An Introduction to Biometric Recognition. IEEE Trans. Circ. Syst. Video Tech. 14(1), 4–20 (2004)
2. Samal, A., Iyengar, P.: Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey. Pattern Recognition 25(1), 65–77 (1992)
3. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face Recognition: A Literature Survey. ACM Computing Surveys 35(4), 399–458 (2003)

4. Bowyer, K.W., Chang, K., Flynn, P.: A Survey of Approaches and Challenges in 3D and Multi-Modal 3D + 2D Face Recognition. Computer Vision and Image Understanding 101(1), 1–15 (2006)
5. Sinha, P., Balas, B., Ostrovsky, Y., Russell, R.: Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About. Proceedings of the IEEE 94(11), 1948–1962 (2006)
6. Zeng, Z., Pantic, M., Rosiman, G.I., Huang, T.S.: A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. IEEE Trans. Pattern Anal. Mach. Intell. 31(1), 39–58 (2009)
7. Craw, I., Costen, N., Kato, T., Akamatsu, S.: How Should We Represent Faces for Automatic Recognition? IEEE Trans. Pattern Anal. Mach. Intell. 21(8), 725–736 (1999)
8. Tanaka, J.W., Farah, M.J.: Parts and Wholes in Face Recognition. Quarterly Journal of Experiment Psychology 46A(2), 225–245 (1993)
9. Tao, D., Tang, X., Li, X.: Which Components Are Important for Interactive Image Search? IEEE Trans. Circ. Syst. Video Tech. 18(1), 3–11 (2008)
10. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience 3(1), 71–86 (1991)
11. Belhumeur, P.N., Hespanda, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 711–720 (1997)
12. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.: Face Recognition Using Laplacianfaces. IEEE Trans. Pattern Anal. Mach. Intell. 27(3), 328–340 (2005)
13. Scholkopf, B., Smola, A., Muller, K.R.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10(5), 1299–1319 (1998)
14. Liu, Q., Huang, R., Lu, H., Ma, S.: Face Recognition Using Kernel Based Fisher Discriminant Analysis. In: 5th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 197–201 (2002)
15. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face Recognition by Independent Component Analysis. IEEE Trans. Neu. Net. 13(6), 1450–1464 (2002)
16. 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 Anal. Mach. Intell. 26(1), 131–137 (2004)
17. Chai, X., Shan, S., Chen, X., Gao, W.: Locally Linear Regression for Pose-invariant Face Recognition. IEEE Trans. Img. Proc. 16(7), 1716–1725 (2007)
18. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust Face Recognition via Sparse Representation. IEEE Trans. Pattern Anal. Mach. Intell. 31(2), 210–227 (2009)
19. Naseem, I., Togneri, R., Bennamoun, M.: Linear Regression for Face Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 32(11), 2106–2112 (2010)
20. Wiskott, L., Fellous, J., Kruger, N., von der Malsburg, C.: Face Recognition by Elastic Bunch Graph Matching. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 775–779 (1997)
21. Albiol, A., Monzo, D., Martin, A., Sastre, J., Albiol, A.: Face Recognition Using HOG-EBGM. Pattern Recognition Letters 29(10), 1537–1543 (2008)
22. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
23. Ahonen, T., Hadid, A., Pietikäinen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 28(12), 2037–2041 (2006)

24. Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-statistical Model for Face Representation and Recognition. In: 10th IEEE International Conference on Computer Vision, vol. 1, pp. 786–791 (2005)
25. Zhang, B., Shan, S., Chen, X., Gao, W.: Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition. IEEE Trans. Img. Proc. 16(1), 57–68 (2007)
26. Deng, C., He, X., Han, J.: SRDA: An Efficient Algorithm for Large-Scale Discriminant Analysis. IEEE Trans. Knowledge Data Eng. 20(1), 1–12 (2008)
27. Georghiades, A., Belhumeur, P., Kriegman, D.: From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. IEEE Trans. Pattern Anal. Mach. Intell. 23(6), 643–660 (2001)
28. Martinez, A.M., Kak, A.C.: PCA versus LDA. IEEE Trans. Pattern Anal. Mach. Intell. 23(2), 228–233 (2001)
29. Ekman, P., Friesen, W.V.: Facial Action Coding System (FACS). Consulting Psychologists Press, Palo Alto (1978)
30. Tuzel, O., Porikli, F., Meer, P.: Region Covariance: A Fast Descriptor for Detection and Classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
31. Pang, Y., Yuan, Y., Li, X.: Gabor-Based Region Covariance Matrices for Face Recognition. IEEE Trans. Circ. Syst. Video Tech. 18(7), 989–993 (2008)
32. Benabdelkader, C., Cutler, R.G., Davis, L.S.: Gait Recognition Using Image Self-similarity. EURASIP Journal on Applied Signal Processing 2004(1), 572–585 (2004)
33. Shechtman, E., Irani, M.: Matching Local Self-Similarities across Images and Videos. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
34. Junejo, I.N., Dexter, E., Laptev, I., Pérez, P.: Cross-View Action Recognition from Temporal Self-Similarities. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 293–306. Springer, Heidelberg (2008)
35. Stanffer, C., Grimson, E.: Similarity Templates for Detection and Recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 221–228 (2001)
36. Savarese, S., Winn, J., Criminisi, A.: Discriminative Object Class Models of Appearance and Shape by Correlatons. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2033–2040 (2006)
37. Eleyan, A., Demirel, H.: Co-Occurrence Based Statistical Approach for Face Recognition. In: 24th International Symposium on Computer and Information Sciences, pp. 611–615 (2009)
38. Watanabe, T., Ito, S., Yokoi, K.: Co-Occurrence Histograms of Oriented Gradients for Pedestrian Detection. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 37–47. Springer, Heidelberg (2009)
39. Yang, M., Zhang, L., Zhang, L., Zhang, D.: Monogenic Binary Pattern (MBP): A Novel Feature Extraction and Representation Model for Face Recognition. In: 20th International Conference on Pattern Recognition, pp. 2680–2683 (2010)

Coding of Dynamic 3D Mesh Model for 3D Video Transmission

Jui-Chiu Chiang, Chun-Hung Chen, and Wen-Nung Lie

Department of Electrical Engineering
National Chung Cheng University, Chia-Yi, 621, Taiwan, ROC
{rachel,ieewnl}@ccu.edu.tw

Abstract. Recently, 3D video has gained increasing attention in multimedia field. The representation of 3D video is often based on dynamic 3D mesh model, which is reconstructed from multi-view video, plus surrounding texture information for rendering, so that arbitrary novel views can be synthesized accordingly. However, the dynamic 3D mesh model herein is not time-consistent, resulting in a difficulty in applying traditional mesh compression tools efficiently (e.g., MPEG-4 AFX 3DMC). In this paper, we modify the 3DMC algorithm for the coding and transmission of 3D video, taking its advantage of high coding efficiency for edge topologies and enhancing it with 3D motion estimation of vertices between two time-successive mesh models. Experiment results show that our method can reach about 30 times of compression ratio. Compared to MPEG-4 AFX 3DMC, under comparable reconstruction quality, our algorithm has a bit rate saving of about 20%~45%.

Keywords: 3D video, 3D mesh, 3D motion estimation.

1 Introduction

Nowadays, the development of multimedia video has already been promoted from 2D to 3D, or from single-view toward multi-views. With 3D video, observers are capable of seeing around an object by changing the viewing directions at their will. The observed views are no longer restricted to those captured by the really arranged cameras. For example, a dancer on the stage can be seamlessly looked around (or evenly zoomed-in/out) by an audience who controls a mouse to change the viewing direction [6]. This effect however relies on image projection of a 3D model or novel-view synthesis from multiply captured images. In a foreseeable future, 3D video will have more applications in education, art, entertainment, etc.

The representation of 3D video can be divided into two kinds: one is multi-view video plus depth information for each view [4], the other is dynamic 3D mesh model plus surrounding texture information for rendering [5]. Both kinds of methods need to arrange a number of inwards cameras for capturing the object views from several discrete directions. The former then estimates the disparity or depth information from any two adjacent views so that arbitrary novel views can be synthesized by using the DIBR (Depth Image-Based Rendering) technique, whereas the latter constructs a 3D

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 144–152, 2011.

model from all the captured views so that arbitrary views can be synthesized by projecting the 3D model onto an image plane, with textures being rendered thereon. Both methods take their own advantages and disadvantages. For example, multi-view video plus depth approach is advantageous of its system simplicity but is difficult in accurate disparity/depth estimation. On the other hand, the 3D mesh approach requires more cameras for accurate 3D model reconstruction, but benefits from flexibility in arbitrary view generation.

This work is a part of a 3D video system that adopts the approach of dynamic 3D mesh and texture rendering, focusing on the compression of the dynamic 3D mesh models that are reconstructed from a number of inwards cameras around the target objects. Though 3D mesh models have ever been widely used, the ones reconstructed in 3D video systems are different from those generated via tools of computer graphics. The most important is that topologies of the 3D mesh models (including the number of vertices, vertices comprising the triangular meshes, and number of triangles) most likely vary from time to time, presenting no correspondences between two 3D mesh models at successive time instants. This case, however, will not happen in computer-graphics-generated 3D mesh models.

The development of 3DMC (3D mesh coding) in MPEG-4 part 16 AFX (Animation Framework eXtension) has been mature for several years. This tool is however mainly developed for a single static model, but not for dynamic models generated/reconstructed at successive time instants. Though some techniques [7, 8] have been proposed to fill this gap, they are based on time-consistent dynamic 3D meshes (i.e., assuming that the vertex and edge sets of the 3D mesh models at successive time instants are kept preserved). As mentioned earlier, this assumption does not hold for 3D video applications. Theoretically, the MPEG-4 AFX tool can be applicable to the above-mentioned 3D video system by individually encoding the 3D mesh model reconstructed at each time instant. The encoding efficiency can be further improved by exploring the relation or redundancies between two time-successive mesh models.

It is the goal of this work to encode time-inconsistent dynamic 3D mesh models for 3D video transmission. The algorithm is essentially a modification of the MPEG-4 AFX, taking advantage of its high coding efficiency for edge topologies and enhancing it with accurate prediction of vertices between two time-successive mesh models.

2 Topological Surgery for 3D Mesh Model

There are many projects in MPEG-4 Part 16 AFX which concern about animation. Among them, 3DMC (3D mesh coding) was targeted for the compression of a single 3D mesh model. It adopts a “geometric compression through topological surgery (TS)” algorithm [1], which keeps the relation between meshes accurately and is capable of reaching a high compression performance.

The manner that 3DMC considers for a single mesh model can be named as "*intra-model*" coding. However, to handle dynamic 3D mesh models, the redundancy between each pair of time-successive mesh models should be taken into account for coding efficiency improvement. We name this "*inter-model*" coding, following the terminologies from the state-of-the-art video coding. The TS technique [1] is to dissect a mesh model into a spanned tree and then perform encoding of the resulting "vertex tree", "triangle tree", and "vertex coordinates". Among the above three quantities to be encoded, we first explore the inter-model redundancies for vertex coordinates in this work. For time-inconsistent dynamic 3D mesh models, the inter-model redundancies for the vertex trees and triangle trees remain still an open issue to the researchers. In one word, our algorithm follows the tree spanning and encoding procedures proposed in 3DMC, but modifies the encoding of vertex coordinates that constitute the most significant part of the resulting bit stream.

3 Inter-model Coding Based on 3D Motion Estimation

We borrow the concept of inter-frame 2D motion estimation from video coding for this inter-model vertex prediction, that is, "*3D motion estimation*" which predicts vertex coordinates of the current (time) model from the previous (time) model and encodes the residuals. It is inefficient for each vertex to have a prediction parameter (e.g., 3D transform parameters) for calculating the residuals. Rather, we group vertices to adopt a limited set of 3D transform parameters. It is observed that transformation between successive (-time) 3D mesh models is often non-rigid (e.g., motions of the human's body and limbs might not be consistent). Hence, k-means clustering algorithm (in this work, k=5, considering human's body and 4 limbs) is adopted to partition the vertices of each model into k groups. Then the well-known ICP (Iterative closest point) algorithm [2] is used to align each group of vertices with those (the whole set) in the previous reconstructed (after decoding) mesh model $\tilde{F}_{n-1}$. The estimated 3D transform parameters by using the ICP algorithm are regarded as the 3D motion parameter by which vertices of each cluster can be nearly aligned (or, closest to) with one of those in $\tilde{F}_{n-1}$ (see Fig.1). After 3D motion estimation, the information need to be recorded and transmitted include the indices of the corresponded vertex in $\tilde{F}_{n-1}$ and the displacement (residuals) between the transformed current vertex and the corresponded vertex, written as (*v_index, x_residue , y_ residue , z_ residue*). After k-means clustering and ICP (i.e., 3D motion estimation), the vertices between two consecutive mesh models will be closer so that we can get much less residuals for encoding.

To further improve the coding efficiency, we introduce two procedures: "spatio-temporal search" and "local index search". Spatio-temporal search means that vertex prediction source can not only be from $\tilde{F}_{n-1}$, but also from a subset of $\tilde{F}_n$ itself. The subset that meets this purpose is limited to those vertices preceding the current vertex in the decoding procedure (i.e., vertices with smaller indices). Those vertices not encoded yet will be excluded from consideration in spatio-temporal search. The search results in the spatial and temporal domains are compared and the one with less coordinate residual is chosen as the final prediction source for encoding.

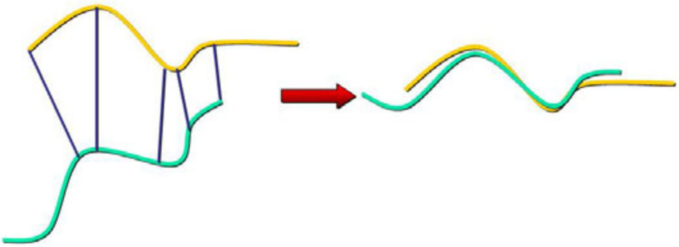


Fig. 1. ICP algorithm [2]

In addition to residuals of vertex coordinates, we still need to encode an extra list that records the indices of the corresponded vertices after 3D motion estimation. “Local index search” plus differential index coding will be beneficial to the coding efficiency of this list. In 3D motion estimation, we need to find a vertex in the previous model that is closest to the transformed current vertex. When the search range is restricted to locally neighboring indices of the prior encoded vertex, both the time complexity and the bit rate required for encoding the referred list can be significantly reduced. An exception is that a full search of $\tilde{F}_{n-1}$ will be still conducted if the residual of vertex coordinates from the local index search result is larger than a given threshold. The flow chart of our modified 3DMC algorithm based on inter-model prediction is summarized in Fig. 2.

4 Experiment Results

The 3D mesh models used in experiments are created from multi-view videos captured from 13 cameras arranged around the targeted objects, as shown in Fig. 3. For multi-view images at each time instant, the visual hull algorithm [3] is applied to reconstruct dynamic 3D mesh models. As mentioned earlier, dynamic 3D mesh models reconstructed in this way will not be time-consistent, that is, the number of vertices and the associated topology information will not be preserved. Frames 0-5 of “robot” (Fig.4) are used for experiments to test our proposed algorithm. (here, a “frame” means a reconstructed 3D mesh model at a time instant)

Table 1 shows the result of compression ratio. Since Frame-0 uses the MPEG-4 AFX-3DMC for encoding, we do not list it in Table 1. It is observed that the average compression ratio is 31.85 for Frames 1-5.

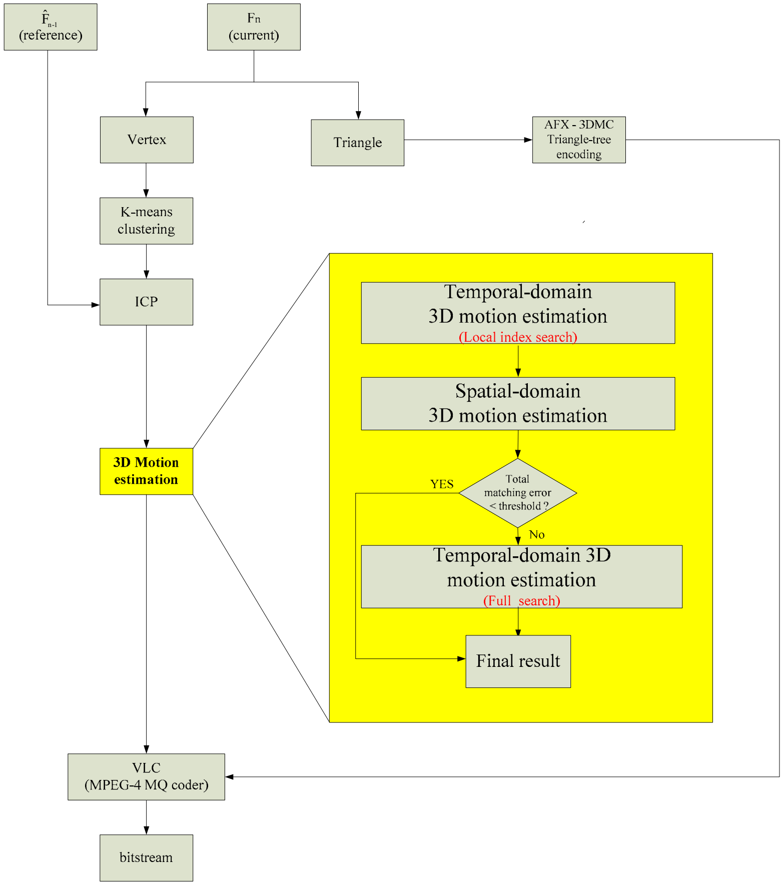

$\hat{F}_{n-1}$ (reference)
Fn (current)
Vertex
Triangle
AFX - 3DMC Triangle-tree encoding
K-means clustering
ICP
3D Motion estimation
Temporal-domain 3D motion estimation
(Local index search)
Spatial-domain 3D motion estimation
Total matching error < threshold ?
YES
No
Temporal-domain 3D motion estimation
(Full search)
Final result
VLC (MPEG-4 MQ coder)
bitstream

Fig. 2. The proposed modified 3DMC algorithm for time-inconsistent dynamic 3D mesh models

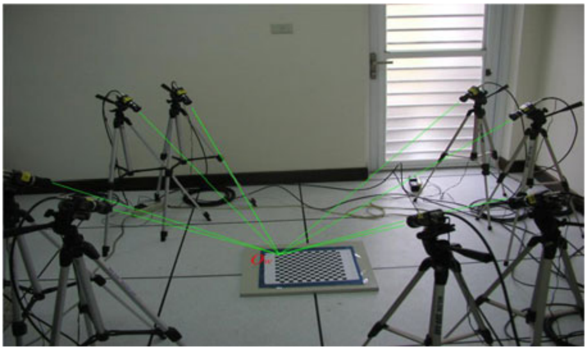


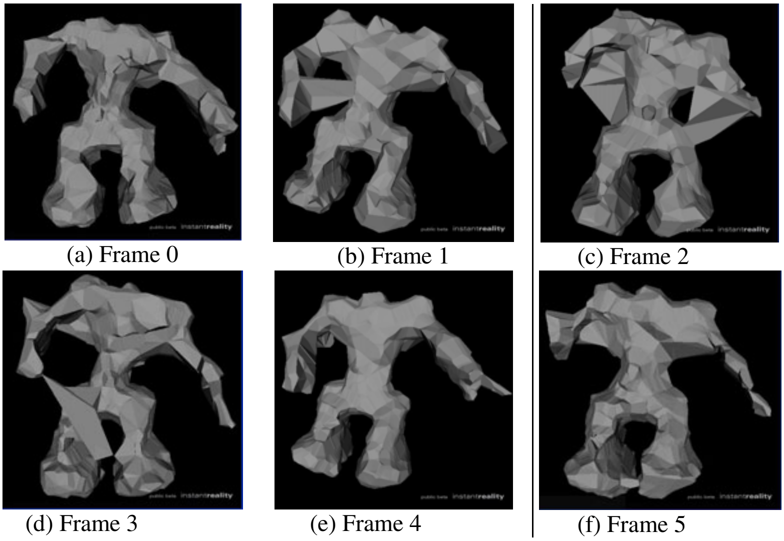
Fig. 3. Part of the multi-view video capturing configuration

Table 1. Compression ratio of our modified 3DMC algorithm

	Original file size (KB)	Compressed file size (KB)	Compression Ratio
Frame1	1339	42	31.88
Frame2	1074	35	30.69
Frame3	1278	40	31.95
Frame4	1323	41	32.27
Frame5	1154	37	31.19

Three quality measures are calculated for the decoded 3D mesh models:

(1) E_1: average norm-1 error (in terms of mm) of vertex position,
(2) E_2: KG error [9] (a well-known measurement in computer graphics),
(3) E_3: SNR (dB) of derived depth image (projecting the 3D mesh model onto a selected image plane to get depth image).



(a) Frame 0
(b) Frame 1
(c) Frame 2
(d) Frame 3
(e) Frame 4
(f) Frame 5

Fig. 4. 3D mesh models reconstructed based on multi-view images

To compare with the MPEG-4 AFX 3DMC, Figs. 5~7 show their R-D curves. Note that each data point of different bit rate is obtained by varying the coordinate accuracy (BPV=8,9,10,12,14 bit per vertex) for MPEG-4 AFX 3DMC or varying residual accuracy (Quality Factor, QF=1,2,5,10, 24, the larger, the more accuracy) for the modified 3DMC.

Figs. 5-7 show that our method outperforms MPEG-4 AFX 3DMC, regarding all three measures. At the same quality, our method has a compression gain of about 20% in bit rate.

We also conduct an experiment on computer-graphics-generated mesh model: "chicken" (Fig.8) to prove the applicability of our modified 3DMC on time-consistent models (but does not take advantage of the vertex correspondence relations). Similarly, we compute the three R-D curves for comparison (not shown here). At a considerable quality, our method outperforms MPEG-4 AFX 3DMC by a bit rate saving of 42.5% (4.6 KB/model vs. 8 KB/model). This better gain lies on the fact that a less noisy topology makes 3D motion estimation more reliable to finding matching vertices between two successive models.

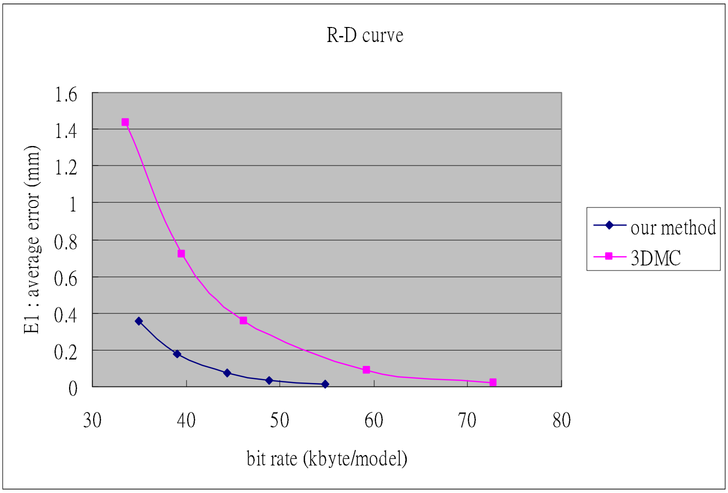


Fig. 5. The R-D curve (E_1 vs. bit rate) in comparison

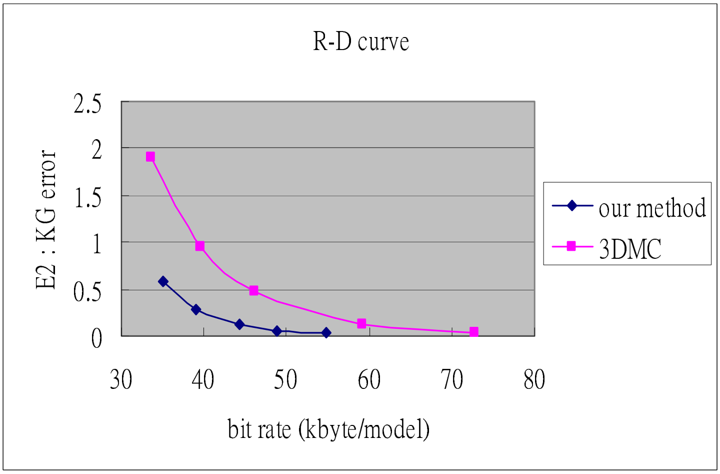


Fig. 6. The R-D curve (E_2 vs. bit rate) in comparison

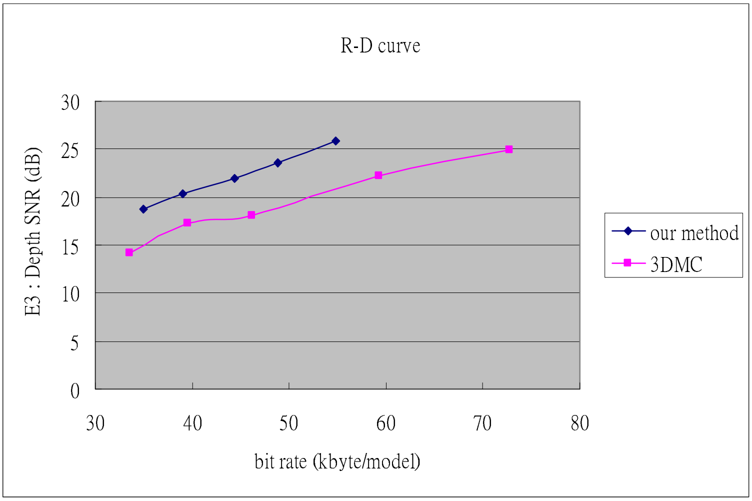


Fig. 7. The R-D curve (E_3 vs. bit rate) in comparison

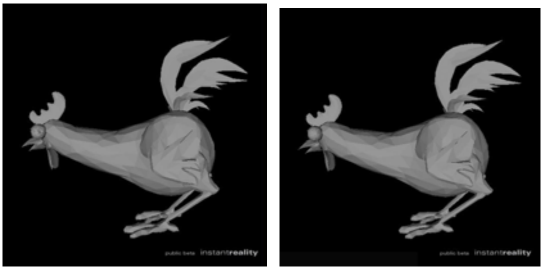


Fig. 8. Frames 0 & 1 of “Chicken” created by computer-graphics tools

5 Concluding Remarks

Essentially, our modified 3DMC algorithm is based on the traditional TS scheme, enhanced with a 3D motion estimation algorithm for vertex prediction between successive models. We also develop two algorithms of spatio-temporal search and local index search to further improve the coding efficiency. The compression ratio depends on the variation (e.g., global behavior or consistency of vertex motions) between two successive models, while that of 3DMC which compresses each model separately is kept less varying if the number of vertices is fixed. Another promising way to further improve coding efficiency is to build vertex correspondence and vertex ordering at the earlier stage of 3D mesh reconstruction, that is, adopting a preprocessing instead of a post-processing (via ICP algorithm).

References

1. Taubin, G., Rossignac, J.: Geometric Compression Through Topological Surgery. ACM Trans. on Graphics 17(2), 84–115 (1998)
2. Besl, P.J., McKay, N.D.: A Method for registeration of 3-D shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
3. Laurentini, A.: The Visual Hull Concept for Silhouette-Based Image Understanding. IEEE Tran. on Pattern Analysis and Machine Intelligence 16(2), 150–162 (1994)
4. Merkle, P., Smolic, A., Muller, K., Wiegand, T.: Multi-View Video Plus Depth Respresentation and Coding. In: Proc. of IEEE Int'l. Conf. on Image Processing (ICIP 2007), vol. 1, pp. I-201 – I-204 (2007)
5. Laurentini, A.: The Visual Hull Concept for Silhouette-Based Image Understanding. IEEE Tran. on Pattern Analysis and Machine Intelligence 16(2), 150–162 (1994)
6. http://www.chiariglione.org/mpeg/technologies/mp-mv/
7. Stefanoski, N., Klie, P., Liu, X., Ostermann, J.: Layered Predictive Coding of Time-Consistent Dynamic 3D Meshes using a Non-Linear Predictor. In: Proc. of IEEE Int'l. Conf. on Image Processing (ICIP 2007), pp. V-109-V-112 (2007)
8. Amjoun, R., Sreaber, W.: Efficient Compression of 3D Dynamic Mesh Sequences. In: Proc. of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, wscg (2007)
9. Karni, Z., Gotsman, C.: Compression of soft-body animation sequences. Computers & Graphics 28(1), 25–34 (2004)

Ray Divergence-Based Bundle Adjustment Conditioning for Multi-view Stereo

Mauricio Hess-Flores[1], Daniel Knoblauch[2], Mark A. Duchaineau[3], Kenneth I. Joy[1], and Falko Kuester[2]

[1] Institute for Data Analysis and Visualization, University of California, Davis, USA
{mhessf,kijoy}@ucdavis.edu
[2] University of California, San Diego, USA
{dknoblau,fkuester}@ucsd.edu
[3] Lawrence Livermore National Laboratory, Livermore, CA, USA
duchaine@llnl.gov

Abstract. An algorithm that shows how ray divergence in multi-view stereo scene reconstruction can be used towards improving bundle adjustment weighting and conditioning is presented. Starting with a set of feature tracks, ray divergence when attempting to compute scene structure for each track is first obtained. Assuming accurate feature matching, ray divergence reveals mainly camera parameter estimation inaccuracies. Due to its smooth variation across neighboring feature tracks, from its histogram a set of weights can be computed that can be used in bundle adjustment to improve its convergence properties. It is proven that this novel weighting scheme results in lower reprojection errors and faster processing times than others such as image feature covariances, making it very suitable in general for applications involving multi-view pose and structure estimation.

Keywords: Multi-view reconstruction, ray divergence, weighted bundle adjustment, confidence ellipsoids, image feature covariances.

1 Introduction

During the past years there has been a surge in the amount of work dealing with multi-view reconstruction of scenes, for industry and in many other modern applications. State-of-the-art algorithms [1] provide very accurate matching, camera poses and scene structure, based on sparse features such as those obtained with the SIFT [2] or related algorithms. These recent algorithms are capable of reconstructing large scenes from even unstructured image sets, obtained for example from the Internet. In such scenarios, camera parameters such as location, orientation and intrinsics may be available or accurately estimated for some of the cameras but not all. This could also be the case even in structured sets of images acquired with the same camera. Because of this reason, despite very accurate feature matching, the accuracy of a multi-view reconstruction still relies on accurate camera parameter calibrations. This creates a great need to identify

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 153–164, 2011.

where and why errors are present in these parameters, specifically without the need to know ground-truth, since this is not always available. In the absence of ground-truth data, multi-view algorithms usually resort to *bundle adjustment* [3] to reduce reprojection error, which is the most meaningful geometric measure of accuracy in the lack of any ground-truth. However, this can be an expensive element in a scene reconstruction pipeline for high numbers of scene points and cameras, despite recent and efficient sparse implementations such as *SBA* [3], and must be used wisely. Furthermore, it requires a good enough starting point close to the global minimum for convergence.

Our main goal in this paper is to show how simple *ray divergence* when attempting scene reconstruction is an inexpensive yet powerful tool that can aid in bundle adjustment convergence for multi-view stereo. Ray divergence is defined as the shortest distance between rays emanating from each respective camera center and through each pixel position of a given feature track, as will be further described in Subsection 2.1. Our work is partially inspired by the recent algorithm of Knoblauch et al. [4], which measures per-correspondence ray divergence when attempting scene reconstruction from a set of initial unconstrained dense correspondences and then decomposes the total error map into errors related to camera parameters and correspondence errors. To the knowledge of the authors there had been no other previous work on such an error factorization without using ground truth knowledge. The ray divergence metric relies on the input feature matches being unconstrained, which is what allows for measuring geometric errors. Using matches generated for example through epipolar geometry-based guided matching would yield no reconstruction error, since these are generated such that they lie on the same epipolar plane with the point they represent in 3D space.

As far as other previous work on camera parameter error analysis, it has been done for the most part with respect to ground-truth values, such as the methodology to test the accuracy of camera pose estimation presented in Rodehorst et al. [5]. The work in Zhao et al. [6] deals with how extrinsic and intrinsic calibration inaccuracies contribute towards depth estimate errors, but for the specific case of a stereo camera pair with a known baseline and other relative positioning information. Benchmarks also exist for reconstruction accuracy [7], though the analysis is done versus ground-truth values, and our algorithm is based on ray divergence rather than the accuracy of exact recovered positions.

In our algorithm, we compute ray divergence per feature track and use it as a joint measure of all camera parameter inaccuracies, without the need for ground-truth knowledge and prior to actually computing the 3D structure. We start out similarly to Knoblauch et al. [4], first computing ray divergences for all available feature matches but with the important difference that we use robust SIFT features instead of dense correspondences, keeping in mind that such feature matches are also unconstrained and therefore it is possible to extract a geometric error unlike in guided matching. We also assume that these feature matches are highly accurate, and this is generally true since sparse SIFT matches are less prone to mismatching due to occlusions, repetitive patterns and texture-less regions than dense correspondences. To further ensure that we have very accurate matches, epipolar geometry-based RANSAC outlier removal [8] is applied prior to computing ray divergences. This in turn allows us to assume that the total ray divergence error corresponds only to camera-related inaccuracies, such that we can avoid the error decomposition in Knoblauch et al. to obtain camera parameter errors.

As will be discussed, the validity of ray divergence as a measure of camera parameter uncertainty can be proven, since it correlates well with Beder et al.'s confidence ellipsoid roundness measure for computed 3D scene points [9] in the case when image feature covariances are set to identity. Furthermore, since ray divergence encodes camera inaccuracy information, we show how it can be used in weighted bundle adjustment to improve its convergence properties. It is shown how this scheme outperforms weighting based on more-expensive image feature covariance metrics [10,11] or Beder et al.'s confidence measure. The entire procedure is first derived for the two-view case, but later shown how this can easily be extended to multiple views. In summary, our algorithm presents a very practical and inexpensive way to measure camera parameter uncertainty in the absence of ground-truth information and use that uncertainty to improve bundle adjustment conditioning. The entire procedure will be described in detail in Section 2, followed by experimental results (Section 3) and conclusions (Section 4).

2 Proposed Algorithm

Our analysis will begin with the two-view case, where it is first shown in Subsection 2.1 how to compute ray divergence, and in Subsection 2.2 how to set up weighted bundle adjustment based on ray divergence values. The extension to multiple views will be outlined in Subsection 2.3.

2.1 Two-View Ray Divergence Calculation

The first step in our algorithm is to compute ray divergence per feature match, similarly to Knoblauch et al. [4], except we start with sparse SIFT features [2] instead of dense correspondences. In the case of *perfect* feature matches, camera intrinsics and extrinsics and no radial distortion, rays starting from each camera center and through the respective image plane feature location should intersect at an exact position in 3D space, but due to any inaccuracies this generally will not occur. We define ray divergence as the shortest distance between such rays, as depicted on the left image of Fig. 1. As mentioned earlier, due to accurate feature matching ray divergence is assumed to correspond entirely to camera parameter inaccuracies, which turns out to be a good approximation even if there are small matching errors. Matches will never be perfect in reality, but we filter bad matches through RANSAC on the epipolar geometry, using a $3.84\sigma^2$ inlier threshold on Sampson error [8].

Ray directions D_i for the two cameras are calculated per Eq. 1, with x_i and y_i being the pixel coordinates in each image. The absolute orientation R_i and

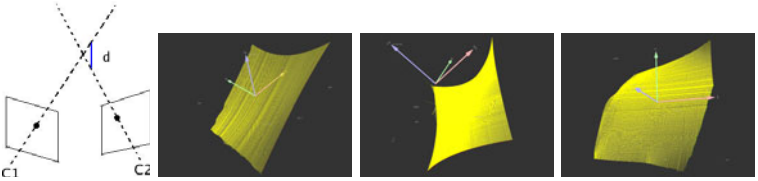


Fig. 1. Concept of ray divergence d (*left*), and sample dense camera parameter error maps for image pairs from different datasets, to depict their smooth variation

position C_i for each of the two cameras is computed by factorizing the *essential matrix*, which can be computed from feature matches using *N-point* algorithms [8]. The cameras' intrinsic parameters (such as focal length and principal point, with no pixel skew) are assumed to be at least roughly known in order to create each 3×3 matrix K_i.

$$D_i = R_i * K_i^{-1} * \begin{pmatrix} x_i \; y_i \; 1 \end{pmatrix}^T \ . \tag{1}$$

Given the camera center locations C_i, the shortest distance between the two rays corresponds to the Euclidean distance between the nearest distance points P_i on each ray as shown in Eq. 2, with t_i defining the distance to move along each ray. Finally, the ray divergence d can be obtained from $d_i = |P_1 - P_2|^2$. This error comprises any inaccuracies with the camera poses, intrinsics or radial distortion, and influences scene reconstruction in a global, smooth manner [4].

$$P_i = C_i + t_i * D_i \ . \tag{2}$$

The ray divergence d is then computed for all available feature matches. In Knoblauch et at. [4], the resulting set of divergences corresponds to the total reconstruction error which is a function of both feature matching errors and camera-related errors, but as mentioned earlier we assume here that the entire error corresponds to the cameras. Therefore, we can say that ray divergence d_i for a given feature match is a function of relative rotation between the two cameras R_{rel}, relative translation T_{rel}, intrinsic parameters for the two cameras K_1 and K_2, and radial distortion, which we'll represent as distorted pixel coordinates (x_{ri}, y_{ri}), such that $d_i = f(R_{rel}, T_{rel}, K_1, K_2, x_{ri}, y_{ri})$.

To show how errors in these parameters affect ray divergence in a global, smooth manner, and for visualization purposes since it becomes more difficult to show using sparse matches, we computed dense correspondences through a standard optical flow method to obtain a total ray divergence map for a few test sequences. Each was factorized into camera-parameter error maps, modelled as smooth B-spline surfaces, and correspondence error maps (remaining high-frequency components). The resulting camera-parameter error maps are shown in Fig. 1. Starting with sparse features, a smooth but sparse set of surface points is obtained as shown in Fig. 2 for the *Palmdale* dataset, which shows grayscale-coded ray divergence values for all available matches. In general, it has been observed that the highest divergences tend to occur towards the edges of images

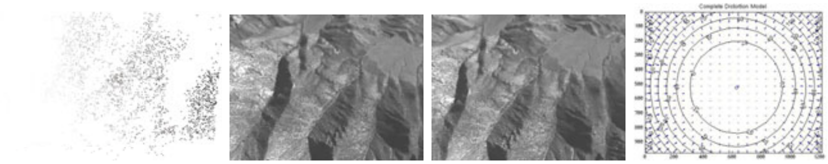


Fig. 2. Ray divergences (*left*) for the set of matches from a pair of *Palmdale* dataset images (*middle*), displayed such that lighter colors indicate higher divergences. The true radial distortion map for the used camera, in pixels, is also displayed (*right*).

(as seen in Fig. 2, where most matches are on the left-hand side of the images) in part because of radial distortion, and it becomes clear that we want such matches to have less of an influence in bundle adjustment because of their higher ray divergence, as discussed further in Subsection 2.2.

2.2 Bundle Adjustment Weighting with Ray Divergences

Now that ray divergences have been computed, and assuming that these are a function mainly of camera parameter inaccuracies, it will be shown how these values can be used as input weights to bundle adjustment in order to improve its convergence properties. However, one further step before applying bundle adjustment is to obtain initial estimates for the scene's structure. We use Lindstrom's triangulation algorithm [12] due to its superior accuracy and speed with respect to standard linear triangulation [8].

Weighted Bundle Adjustment. The objective of bundle adjustment is to adjust pose and structure estimates in such a way that the total reprojection error of the 3D points with respect to their corresponding 2D feature track positions in each camera is minimized [8]. The cost function which is traditionally minimized can be expressed as the sum of squares of reprojection errors between each 3D point and the feature matches which yielded it, as shown in Eq. 3 for the general case of N 3D points seen in M cameras.

$$min(a_j, b_i) \sum_{i=1}^{N} \sum_{j=1}^{M} v_{ij} (d(Q(a_j, b_i), x_{ij}))^2 \ . \tag{3}$$

Here, x_{ij} is the position of the i_{th} feature on image j. The binary variable v_{ij} equals '1' if point i is visible in image j ('0' otherwise). The vectors a_j and b_i parametrize each camera j and 3D point i, respectively, with $Q(a_j, b_i)$ as the reprojection of point i on image j. Finally, d^2 is the Euclidean distance in each image between each original correspondence and its associated reprojection. This minimization involves a total of $3N + 11M$ parameters, and can be achieved using the Levenberg-Marquardt algorithm. The *SBA* implementation [3] was used, since it exploits the sparse block structure of the normal equations solved at each iteration to greatly speed up the process.

The Levenberg-Marquardt algorithm is based on solving the *augmented normal equations* at each iteration. In *weighted* bundle adjustment, each input feature is weighted differently with the objective of improving convergence by giving less weight to those features that are more likely to be inaccurate. In practice, these weights are implemented as covariances. The normal equations have the form shown in Eq. 4, but when using weighted bundle adjustment, the equations change to the form shown in Eq. 5, where Σ corresponds to a block-diagonal matrix consisting of 2×2 covariance matrices for each input feature, J is the parameter Jacobian matrix, δ_p the parameter update step, μ the damping term and ϵ the error vector.

$$(J^T J + \mu I)\delta_p = J^T \epsilon \; . \tag{4}$$

$$(J^T \Sigma_x^{-1} J + \mu I)\delta_p = J^T \Sigma_x^{-1} \epsilon \; . \tag{5}$$

Comparison with Reconstructed Point Confidence Ellipsoid Roundness. Before proceeding, we wish to analyze the validity of ray divergence as a measure of camera errors, such that it can aid in bundle adjustment. Beder et al. [9] present an algorithm to determine the best initial pair for a multi-view reconstruction. Their analysis is based on computing a confidence ellipsoid for each computed 3D scene point X, such that its roundness measures the quality of each obtained point. For two views, the covariance matrices of image feature matches x' and x" are given by C' and C" respectively [10,11]. Then, the covariance matrix C_{XX} of the distribution of the scene point coordinates X is proportional to the upper left 4×4 submatrix $N^{-1}_{1:4,1:4}$ for the inverse of the 5×5 matrix N given by Eq. 6. The A and B matrices encode information related to the projection matrices for the two cameras, the image coordinates of the feature match yielding the scene point, and the 3D point coordinates.

$$N = \begin{pmatrix} A^T \left(B \begin{pmatrix} \text{C'} & 0 \\ 0 & \text{C"} \end{pmatrix} B^T \right)^{-1} A & X \\ X^T & 0 \end{pmatrix} \; . \tag{6}$$

Now, if the homogeneous vector $X = [X_0^T, X_h]^T$ is normalized to Euclidean coordinates, the covariance matrix of the distribution of the Euclidean coordinates is given by Eq. 7, where J_e corresponds to the Jacobian of a division of X_0 by X_h.

$$C^{(e)} = J_e C_{XX} J_e^T \; . \tag{7}$$

Finally, if we perform the singular value decomposition of the matrix $C^{(e)}$, the roundness R of the confidence ellipsoid is obtained as the square root of the quotient of the smallest singular value λ_3 and the largest singular value λ_1, per $R = \sqrt{\frac{\lambda_3}{\lambda_1}}$. The value of R lies between 0 and 1, and only depends on the relative geometry of the two poses, the feature positions and the 3D point; radial distortion is not modelled.

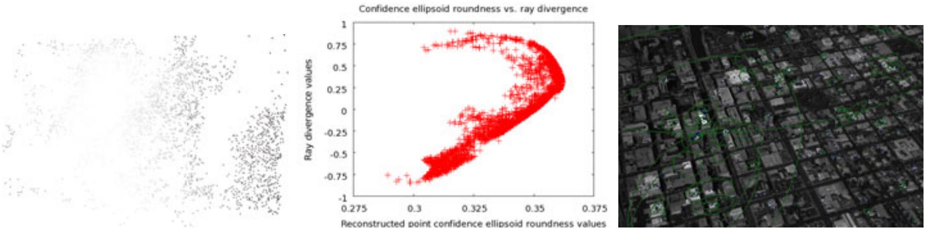


Fig. 3. Reconstructed point confidence ellipsoid roundness values using identity image feature covariances (*left*) for the set of matches from a pair of *Palmdale* dataset images, where lighter colors indicate lower roundness values. The middle image shows its correlation with ray divergence. The right image displays Zeisl's covariance metric values [11] for SIFT features in a *Stockton* image as green ellipsoids.

Something very important to note here is that image feature covariances [10,11] are defined completely by the intensity variations in local neighborhoods and thus may look rather random to visual inspection, with no clear pattern as the image is traversed, as seen on the right in Fig. 3. On the other hand, the surface of ray divergences has a much smoother shape, which is a function of all camera parameter inaccuracies. So if we filter out all features that have high image covariances, matches obtained between remaining 'good' features are still bound to the information ray divergence provides, in order to know if they're overall good or bad matches for reconstruction purposes. This is the power of using ray divergence to weight bundle adjustment, since it provides information beyond just the feature matching uncertainty. For example, two *perfect* matches could still yield a non-zero ray divergence due to camera inaccuracies. Therefore, using ray divergence or even the values provided by Beder et al.'s metric [9], though more expensive to compute and not inclusive of radial distortion, provide a stronger constraint towards weighting bundle adjustment than image-based covariances [10,11]. The right side of Fig. 3 shows the result of applying Zeisl's image covariance metric [11] on a select group of SIFT features, displayed as ellipses with size proportional to covariance values. The left side shows the smooth transitions in values for Beder et al.'s confidence ellipsoid roundness [9] using identity image feature covariances, and the middle shows its correlation with ray divergence. Though it is not an exact correlation because of differences near the edges of images, where the behavior is slightly different, the bulk of points show a very good correlation (a coefficient of 0.93 for the main linear part of this particular plot), such that higher divergences, in absolute value, exhibit lower roundness.

Gaussian Weighting. A close look at a ray divergence histogram reveals a smooth curve, typically reaching a maximum near zero. If we assume that the probability $p(d)$ that a given feature match exhibits a ray divergence d is given by Eq. 8, where μ_d corresponds to the mean ray divergence and σ_d to its standard deviation for a given two-view set of feature matches, we can essentially assume that ray divergence histogram values follow a Gaussian probability density function (pdf) and use these values as weights for bundle adjustment. The average and standard deviation are computed directly from the ray divergences for the available set of feature matches. Since these weights must be input as 2×2 covariance matrices, we assume an isotropic probability distribution and set the diagonal elements with equal pdf-based values, while setting the remaining two elements to zero. It is very important to note that we want to penalize low pdf values since these correspond typically to higher divergences. Therefore, we 'invert' the pdf values and place this number along the diagonal; their original values are obtained again later from matrix inversion while solving the augmented weighted normal equations. This results in higher covariances providing lower weights.

$$p(d) = \frac{1}{2\pi\sigma_d^2} e^{\frac{|d-\mu_d|^2}{2\sigma_d^2}} \quad . \tag{8}$$

The advantages of using Gaussian values as weights is that positive weights are always obtained, no matter what the divergence values are or if they show zero-crossings. The area under the computed Gaussian curve is always unity, by definition, and this is helpful towards mathematical stability since very large variations between the smallest and largest assigned weights is not typical. Also, exponentials are much cheaper to compute than for example a singular value decomposition, as needed in Beder et al.'s algorithm [9]. Finally, ray divergence transitions are smooth such that high ray divergences should be assigned higher covariances than lower ones.

2.3 Extension to Multiple Views

The extension to multiple views is rather simple, and is based directly on the two-view case. In a sequential multi-view pipeline, since covariances have to be specified as 2×2 covariance matrices for each feature of a given feature track, for each feature in a new image we simply assign the Gaussian-based weight corresponding to the ray divergence for the feature's match to the prior image. Average and standard deviation are obtained from the set of pairwise matches between the two most recent images, in order to compute the pdf prior to computing each individual weight. Covariances for the features in the very first image can be initialized to identity, or by computing them from images [10,11] for better initial accuracy. This way of chaining pairwise consecutive estimates works well no matter what the number of frames as long as pairwise ray divergence estimates are well-conditioned, which can usually be achieved through a prior *frame decimation* [13]. An analysis of this baseline effect on divergences is shown in Section 3. For non-sequential cases, the average of all ray divergence values for all matches to a given feature could potentially be used, though we have yet to test this case.

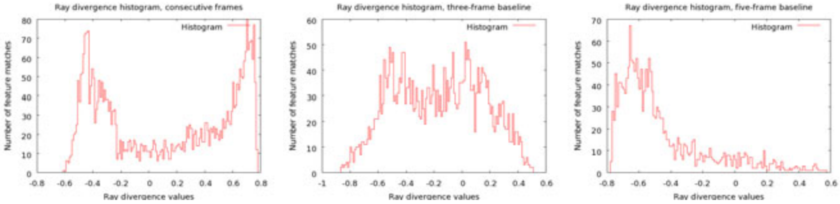


Fig. 4. Ray divergence histograms at increasing baselines (*left to right*), for pairwise frames from the *Stockton* dataset

3 Results

The algorithm was tested on real scenes such as *Stockton*, *Palmdale*, *castle-P19* [7] and *Medusa* [14], as well as synthetic scenes such as *Megascene1* and *Coneland*. All tests were conducted on a single-core *Intel Xeon* machine at 2.80GHz with 1 GB of RAM, on one thread. For all tests, we assume that the same camera is used per dataset and have initial values available for the focal length and principal point, though these in some cases were inaccurate. Images were not undistorted prior to testing, and were acquired sequentially.

One important initial experiment consisted in analyzing the behavior of ray divergence given different baselines. For this, we started out with one frame of the *Stockton* sequence and then obtained ray divergences at different baselines from that particular frame. In Fig. 4, results show that Gaussian fitting works well for 'good' baselines, which are typically achieved by applying frame decimation [13] or other choosing algorithms [1] such that the baseline is not too small for linear triangulation but not too small or large for pose estimation degeneracies to occur. This was also verified in several other datasets. The middle image shows the most smooth histogram, and that is where frame decimation picked the best keyframe. In general, with good baselines ray divergence histograms are smooth and can generally be approximated well by Gaussian fitting. With other baselines, ray divergences would not be suitable for Gaussian fitting and for bundle adjustment, since the values are more heavily affected by noise. A good frame decimation is key to our algorithm's success. Table 1 shows the reprojection error and processing time results for these different baselines, where it is shown that the frame decimation keyframe yielded the lowest reprojection error and processing time per point.

In the next experiment, we compared processing times and reprojection errors obtained using weighted bundle adjustment under four different conditions: bundle adjustment weighted by image feature covariances [10], by confidence ellipsoid roundness with and without including image feature covariances, and based on ray divergences. This was only performed on *good* two-view baselines, obtained with prior frame decimation. Table 2 shows the results for some test datasets. Average values for all test parameters were obtained across pairwise frame analysis for all consecutive pairs of each dataset. Unweighted bundle adjustment was not compared, since the comparison would not be direct. Time

Table 1. Number of points, final total reprojection error R (pixels), bundle adjustment iterations I, processing time t in seconds and min/max ray divergence for Gaussian-pdf ray divergence-weighted bundle adjustment at different baselines, for the *Stockton* dataset. Best results were obtained for the three-frame frame decimation keyframe.

Baseline	Points	R	I	t	min_d	max_d
Consecutive	3605	0.049	150	4.24	−0.606	0.774
3 frames	3369	0.013	33	0.83	−0.863	0.508
5 frames	1831	0.200	73	0.87	−0.774	0.561
8 frames	476	0.111	30	0.09	−0.297	0.537

Table 2. Iterations I, final total reprojection error R (pixels) and processing time t (seconds) in (I, R, t) format obtained using bundle adjustment under four different weighting schemes: image feature covariances (CBA), reconstructed point confidence ellipsoid roundness with ($UWBA$) and without including image feature covariances ($UIBA$), and Gaussian-pdf with ray divergences ($RDBA$)

Dataset	CBA	$UWBA$	$UIBA$	$RDBA$
Stockton	$43, 0.621, 0.90$	$40, 0.171, 0.84$	$37, 0.072, 0.79$	$38, 0.015, 0.78$
Palmdale	$23, 4.687, 0.45$	$22, 1.692, 0.38$	$20, 0.831, 0.41$	$22, 0.113, 0.37$
castle-P19	$150, 281.13, 0.99$	$150, 4150, 0.95$	$150, 1046.1, 0.88$	$97, 90.036, 0.62$
Dinosaur	$26, 2.631, 0.06$	$22, 0.286, 0.05$	$24, 0.09, 0.05$	$24, 0.162, 0.05$
Megascene1	$49, 12.14, 0.04$	$42, 0.179, 0.03$	$45, 0.074, 0.03$	$46, 0.124, 0.04$
Coneland	$150, 28052, 1.10$	$150, 1880.38, 0.99$	$115, 599.88, 0.79$	$126, 81.86, 0.90$

is consumed by the *SBA* software [3] to read-in covariance data, and there is matrix inversion for covariance matrices and multiplication of these with Jacobian matrix elements at each iteration, so processing times are typically higher when using covariances. Even so, our bundle adjustment weighting outperforms unweighted bundle adjustment as far as final reprojection error in almost every case, as seen on the right in Fig. 6 where NBA represents the unweighted case. It can be seen that ray divergence-based weighting outperforms every other type of weighting in just about every category, though it's slightly slower and with a higher reprojection error than the more-expensive $UIBA$ in a few cases. Overall, our weighting scheme provides the best combination of processing time, final reprojection error and computational complexity in computing weights. As far as complexity, Beder's algorithm ($UWBA$ and $UIBA$) for example includes the inversion of a 5×5 matrix and two singular value decompositions of a 4×4 and a 3×3 matrix, whereas ray divergence computation does not involve SVD or inversions at all. The feature covariance method CBA is also more expensive, requiring multiple exponential evaluations for each covariance matrix, whereas our method computes a single exponential value.

Having proven that the algorithm performs very well on pairwise reconstructions, it was applied as explained in Subsection 2.3 to perform multi-view reconstructions using our sparse multi-view reconstruction pipeline. Fig. 5 shows on

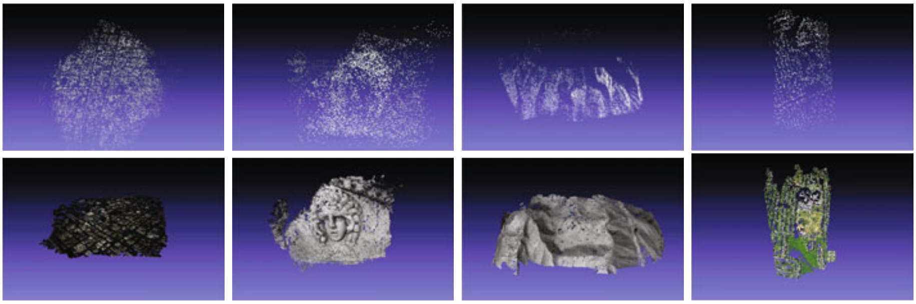


Fig. 5. Top row: sparse multi-view reconstructions for the *Stockton* (*left*), *Medusa* (*middle left*), *Palmdale* (*middle right*) and *Megascene1* (*right*) datasets. Their respective dense reconstructions using the PMVS algorithm [15] are shown on the bottom.

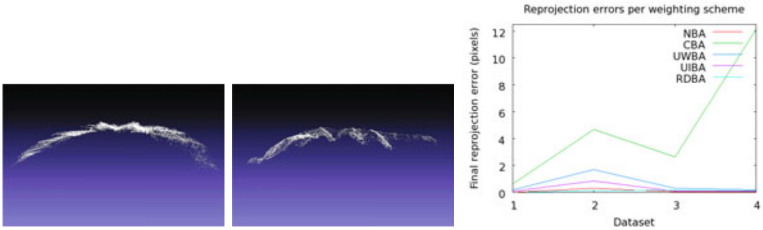


Fig. 6. Side view of a multi-view reconstruction showing the effect of using distorted images (*left*) versus images undistorted with parameters recovered per our algorithm (*middle*), for the *Palmdale* dataset. Total reprojection errors are lower than with other weighting schemes (*right*), as shown for a few datasets.

the top row sparse reconstructions that were obtained while applying sequential multi-view reconstruction, bundle-adjusting with each added image using ray divergence-based weighting. These high-quality sparse reconstructions allow for other algorithms to be applied, such as dense reconstructions with the PMVS algorithm [15] as shown on the bottom row of Fig. 5. Fig. 6 shows the effect on scene reconstruction of using original distorted images versus versions that were undistorted using parameters recovered with our weighted bundle adjustment.

4 Conclusions

An algorithm that makes use of scene reconstruction ray divergence for weighting bundle adjustment and improving its convergence properties was introduced. It was shown that ray divergence, which is a function of all camera parameter inaccuracies, is more efficient to compute and outperforms other weighting schemes such as those based on image feature covariances. There is no dependence on ground-truth information, and results show an improved convergence on different real and synthetic scene types.

Acknowledgments. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

1. Snavely, N., Seitz, S.M., Szeliski, R.: Photo Tourism: Exploring Photo Collections in 3D. In: SIGGRAPH 2006: ACM SIGGRAPH 2006 Papers, pp. 835–846. ACM, New York (2006)
2. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal On Computer Vision 60, 91–110 (2004)
3. Lourakis, M., Argyros, A.: The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece (2000)
4. Knoblauch, D., Hess-Flores, M., Duchaineau, M., Kuester, F.: Factorization of Correspondence and Camera Error for Unconstrained Dense Correspondence Applications. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Wang, J.-X., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5875, pp. 720–729. Springer, Heidelberg (2009)
5. Rodehorst, V., Heinrichs, M., Hellwich, O.: Evaluation of Relative Pose Estimation Methods for Multi-Camera Setups. In: International Archives of Photogrammetry and Remote Sensing (ISPRS 2008), Beijing, China, pp. 135–140 (2008)
6. Zhao, W., Nandhakumar, N.: Effects of Camera Alignment Errors on Stereoscopic Depth Estimates. Pattern Recognition 29, 2115–2126 (1996)
7. Strecha, C., von Hansen, W., Gool, L.J.V., Fua, P., Thoennessen, U.: On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In: CVPR 2008, pp. 1–8 (2008)
8. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press (2004)
9. Beder, C., Steffen, R.: Determining an Initial Image Pair for Fixing the Scale of a 3D Reconstruction From an Image Sequence. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 657–666. Springer, Heidelberg (2006)
10. Brooks, M.J., Chojnacki, W., Gawley, D., van den Hengel, A.: What Value Covariance Information in Estimating Vision Parameters? In: ICCV 2001, pp. 302–308 (2001)
11. Zeisl, B., Georgel, P., Schweiger, F., Steinbach, E., Navab, N.: Estimation of Location Uncertainty for Scale Invariant Features Points. In: BMVC 2009, pp. xx–yy (2009)
12. Lindstrom, P.: Triangulation Made Easy. In: CVPR, 1554–1561 (2010)
13. Knoblauch, D., Hess-Flores, M., Duchaineau, M., Joy, K.I., Kuester, F.: Non-Parametric Sequential Frame Decimation in Low-Memory Streaming Environments. In: 7th International Symposium on Visual Computing, Las Vegas, Nevada, pp. 363–374 (2011)
14. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual Modeling with a Hand-Held Camera. International Journal of Computer Vision 59, 207–232 (2004)
15. Furukawa, Y., Ponce, J.: Accurate, Dense, and Robust Multi-View Stereopsis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)

Temporally Consistent Disparity and Optical Flow via Efficient Spatio-temporal Filtering

Asmaa Hosni[*], Christoph Rhemann[**], Michael Bleyer, and Margrit Gelautz

Institute for Software Technology and Interactive Systems
Vienna University of Technology, Vienna, Austria
{asmaa,rhemann,bleyer,gelautz}@ims.tuwien.ac.at
http://www.ims.tuwien.ac.at

Abstract. This paper presents a new efficient algorithm for computing temporally consistent disparity maps from video footage. Our method is motivated by recent work [1] that achieves high quality stereo results by smoothing disparity costs with a fast edge-preserving filter. This previous approach was designed to work with single static image pairs and does not maintain temporal coherency of disparity maps when applied to video streams.

The main contribution of our work is to transfer this concept to the spatio-temporal domain in order to efficiently achieve temporally consistent disparity maps, where disparity changes are aligned with spatio-temporal edges of the video sequence. We further show that our method can be used as spatio-temporal regularizer for optical flow estimation. Our approach can be implemented efficiently, achieving real-time results for stereo matching. Quantitative and qualitative results demonstrate that our approach (i) considerably improves over frame-by-frame methods for both stereo and optical flow; and (ii) outperforms the state-of-the-art for local space-time stereo approaches.

1 Introduction

Computing disparity maps from two static images is a well-studied problem and many methods have reported impressive results in the past (see e.g. [7] for an overview). However, applying even the best of these methods to *sequences* of stereo pairs in a frame-by-frame manner, often results in temporally inconsistent disparity maps. This temporal inconsistency is perceived as unpleasing flickering, when using such disparity maps to visualize a video on auto-stereoscopic monitors. Such displays use disparity maps to render novel views from virtual view points.

The goal of this work is to improve the quality of the reconstruction by additionally exploiting the temporal information available in the input sequence. To understand the basic idea behind our spatio-temporal stereo method, let us first

[*] Asmaa Hosni was supported by the Vienna PhD School of Informatics.

[**] Christoph Rhemann and Michael Bleyer received financial support from Vienna Science and Technology Fund (WWTF) under project ICT08-019.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 165–177, 2011.

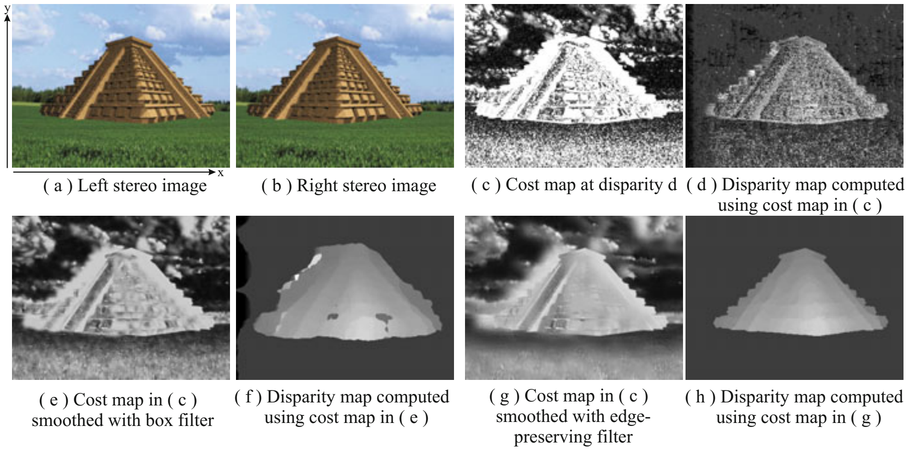

y
x
(a) Left stereo image
(b) Right stereo image
(c) Cost map at disparity d
(d) Disparity map computed using cost map in (c)
(e) Cost map in (c) smoothed with box filter
(f) Disparity map computed using cost map in (e)
(g) Cost map in (c) smoothed with edge-preserving filter
(h) Disparity map computed using cost map in (g)

Fig. 1. Single-frame stereo matching. From a stereo pair, captured at a single moment (a),(b) the associated cost maps and computed disparity maps are shown for different filtering techniques (c)-(h). See the text for a detailed explanation.

consider a conventional local stereo approach that computes a disparity map from a static image pair. Such methods typically build upon a Winner-Takes-All framework [7] and first compute the costs for choosing disparity d at each pixel. An example of the resulting cost map for a single disparity level is shown in fig. 1(c). By choosing for each pixel the disparity with the lowest cost gives a noisy disparity map (see fig. 1(d)). The key idea of block matching algorithms is to improve the result by aggregating the costs over a (square) support window. It is known (see e.g. [1],[7]) that this cost aggregation is equivalent to filtering the cost map with a box filter. Fig. 1(e) shows the cost map in fig. 1(c) after smoothing with a box filter. The resulting disparity map in fig. 1(f) is smoother in comparison to the one generated from the raw cost map in fig. 1(d). The implicit assumption of box filtering methods is that all pixels inside the filter kernel have constant disparity. This assumption is violated if the filter kernel overlaps a depth discontinuity, which often coincides with object boundaries. Therefore, the disparity map generated by box filtering (fig. 1(f)) does not preserve the disparity discontinuities well. A solution is provided by adaptive support weight approaches [2],[3],[6],[1] that locally adjust the filter kernel such that it does not overlap the object boundaries. In particular, such approaches smooth the costs with a *weighted* box filter. The weights are chosen such that they are high in regions which are close in color and distance to the central pixel of the filter kernel and low otherwise. The disparity map in fig. 1(h) better preserves the disparity discontinuities, because it was generated from the cost maps filtered with a weighted box filter (fig. 1(g)).

In this work, we transfer the adaptive support weight concept to the spatio-temporal domain: Given multiple frames in time, a spatio-temporal cost volume is generated by stacking the cost maps of the input frames shown in fig. 2(a) (the cost maps are not visualized in fig. 2). A simple approach would smooth

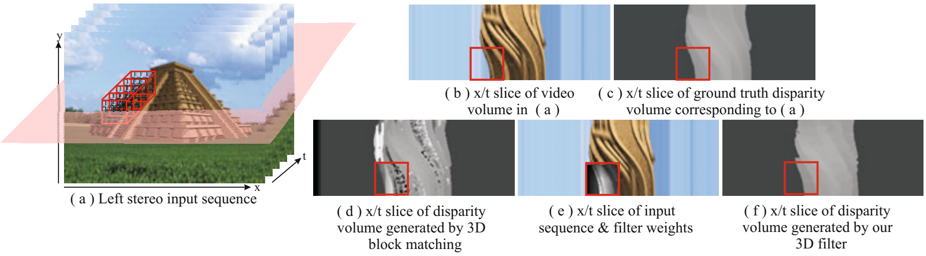

y
x
t
(a) Left stereo input sequence
(b) x/t slice of video volume in (a)
(c) x/t slice of ground truth disparity volume corresponding to (a)
(d) x/t slice of disparity volume generated by 3D block matching
(e) x/t slice of input sequence & filter weights
(f) x/t slice of disparity volume generated by our 3D filter

Fig. 2. Spatio-temporal stereo matching. Given a sequence of stereo images (a) the cost volume is smoothed using a 3D box filter illustrated in red in (a) and (b). The resulting disparity map (d) does not preserve disparity discontinuities. Our approach weights the pixels inside the 3D box filter (e) to achieve a result (f) that is aligned with the space-time object boundaries.

the cost volume with a 3D spatio-temporal box filter as illustrated by the red cube in fig. 2(a). This approach assumes that disparities are constant inside this space-time window. This assumption is not met for the red box in fig. 2(b), since the space-time window overlaps an object boundary. Hence, the resulting disparity map (fig. 2(d)) over-smoothes the spatio-temporal object boundaries. In our approach, we weight the pixels inside the 3D filter kernel - pixels which belong to the same object as the center pixel receive high weights and low weights if they lie on a different object. Our assumption is that the disparity of an *object* is approximately constant over a small space-time window. The filter weights for the kernel outlined in red in fig. 2(e) are visualized by the intensity values inside the red box: Bright pixels encode high weights and dark pixels encode low weights. We see that the weights nicely adapt to the object outline. As a consequence the object boundaries are well preserved in the resulting disparity map shown in fig. 2(f).

The works most closely related to our approach are the methods of [6] and [1]. The stereo method of [6] uses a fast approximation of the bilateral filter to smooth the disparity costs in the spatio-temporal domain. This method is fast (real-time) but the results cannot compete with the state-of-the-art in stereo matching. [1] showed that state-of-the-art results can be achieved at run times similar to [6], by replacing the bilateral filter with the guided image filter [5]. However, the authors of [1] did not adopt their method to temporal sequences. Therefore, the main contribution of our work is to extend the approach of [1] to the spatio-temporal domain. In particular, we present a space-time stereo method that works in real-time and achieves results that outperform the space-time stereo approach of [6]. This is done by extending the guided image filter [5] to the spatio-temporal domain. Furthermore, we leverage this concept to optical flow estimation to achieve temporally smooth flow fields.

Related Work

In the following, we discuss related work that incorporates temporal smoothness constraints for stereo and optical flow estimation.

Although first attempts to enforce temporal coherence for stereo matching go back as far as [20], relatively little research has been conducted in this domain since then. Many previous approaches simply encourage temporally constant disparity solutions. This is either enforced locally by smoothing the cost volume with a rectangular spatio-temporal support window [19],[18],[17] (as discussed above) or globally by propagating the disparity over temporally consecutive pixels (see e.g. [21],[9]). These methods cannot cope well with considerably large movement of scene objects. To account for faster moving scene objects, previous work proposed to orient the rectangular averaging window in the space-time domain such that linear motions can be handled [17]. Other methods compute the optical flow field between consecutive frames and then smooth the disparity values along the computed flow vectors [22],[23]. In contrast, our method implicitly finds the spatio-temporal neighborhood and thus avoids computing a flow field explicitly. Another approach to overcome the problem of flow estimation is [25], where filter responses that capture the local spatio-temporal structure of the video volume are used as matching primitives. A related field of research aims to compute the three-dimensional scene flow [8], which is the computation of the 3D motion field using scene structure information. In contrast, our work aims to reconstruct a temporally smooth disparity map without recovering three-dimensional flow vectors. Note that avoiding the need to compute optical flow is a considerable advantage. We do need to address the optical flow problem that might due to its large label space (consisting of all 2D vectors) be more challenging than the problem that we are actually trying to solve, i.e. temporal stereo. Obviously, we can also avoid the computational overhead that goes along with optical flow computation.

Analogously to space-time stereo matching, only small amount of works have been devoted to temporal consistent optical flow computation. Most algorithms encourage temporal neighboring pixels to be assigned to the same flow vector [16],[15],[14],[13],[24], which however incorporates the assumption that the flow field remains piecewise constant over time. In [12] the less restrictive assumption of constantly moving objects is encoded by encouraging matching pixels in two consecutive frames to take the same flow vector. Our approach is more related to non-local smoothness terms that assume that the flow vector at a certain pixel is similar to the vectors at self-similar pixels in a larger neighborhood. To the best of our knowledge such non-local smoothness terms have only been applied in the spatial domain yet [11], [10].

2 Proposed Algorithm

In this section we describe our algorithm for generating temporally coherent disparity maps. Section 3 adopts this method as temporal regularizer for optical flow estimation.

Our temporal stereo matching algorithm comprises three major steps: (i) construct a spatio-temporal cost volume for each disparity d; (ii) smooth each of these cost volumes with a spatio-temporal filter; and (iii) select for each pixel in the spatio-temporal volume the disparity which holds the lowest costs in its corresponding cost volume. We now discuss each of these steps in detail.

2.1 Cost Volume Construction

In the first step, a cost volume C^d is constructed for each disparity d. This cost volume is a three-dimensional (spatio-temporal) array with axes x, y and t. It stores the costs for choosing disparity d at each voxel $i = (x, y, t)$ (we denote a pixel in a spatio-temporal volume as voxel), where x, y and t are its spatial and temporal coordinates, respectively.

Let I^l and I^r be two spatio-temporal video volumes with axes x, y, t that define a sequence of rectified stereo pairs. Also, let $u = (d, 0, 0)$ be a vector that defines the displacement in x, y and t dimensions. The costs are defined by the correlation of voxel i in I^l with its matching voxel in I^r shifted by vector u:

$$C_i^d = \alpha \cdot M_{i,u}^c + (1 - \alpha) \cdot M_{i,u}^g. \tag{1}$$

Similar to [1], the correlation costs in eq. (1) comprise a color term M^c that is weighted against a gradient term M^g by a factor α. The color term is defined as:

$$M_{i,u}^c = \min\left(\|I_i^l - I_{i-u}^r\|, \tau_c\right),$$

where τ_c is a truncation value and $\|I_i^l - I_{i-u}^r\|$ is the summed-up absolute differences in RGB values. Similarly, we define the gradient term as:

$$M_{i,u}^g = \min\left(\|\nabla_x I_i^l - \nabla_x I_{i-u}^r\|, \tau_g\right),$$

where ∇_x denotes the gradient in x direction and τ_g is a truncation value.

2.2 Spatio-temporal Cost Volume Smoothing

Once the cost volume is constructed for each disparity, we filter each cost volume in the spatio-temporal domain. In particular, the smoothed cost value $\widehat{C}_i^d$, associated with disparity d at voxel i is a weighted average of neighboring voxels in C^d:

$$\widehat{C_i^d} = \sum_j W_{i,j}(I^l) C_j^d. \tag{2}$$

The filter weights $W_{i,j}$ depend on the the sequence of input reference frames I^l and have to be chosen such that spatio-temporal edges in I^l are preserved in the filtered output. A possible choice is to use the weights of the bilateral filter. They have the drawback that an exact implementation is slow and approximations come at the loss of quality. Therefore, we follow [1] where the weights of the recently proposed guided image filter [5] have been used. The guided filter has edge-preserving properties similar to the bilateral filter and its exact implementation has a runtime independent of the filter size. Originally, the guided filter [5] has been defined for the spatial domain only. In this work we extend it to a spatio-temporal kernel, where the filter weights are defined as:

$$W_{i,j} = \frac{1}{|\omega|^2} \sum_{k:(i,j)\in\omega_k} (1 + (I_i^l - \mu_k)^T (\Sigma_k + \epsilon U)^{-1} (I_j^l - \mu_k)). \tag{3}$$

Here, Σ_k and μ_k are the covariance and mean vector computed over the spatio-temporal window ω_k with dimensions $w_x \times w_y \times w_t$, centered at voxel k in the video volume I^l. The number of pixels in this 3D window is denoted by $|\omega|$ and ϵ is a smoothness parameter. I_i^l, I_j^l and μ_k are 3×1 (color) vectors, and the covariance matrix Σ_k and identity matrix U are of size 3×3.

The guided filter weights do not have to be computed explicitly and can be implemented exactly by applying a sequence of linear operations to the input cost volume [5]:

$$\begin{aligned} a_k &= (\Sigma_k + \epsilon U)^{-1} \left(\frac{1}{|\omega|} \sum_{i\in\omega_k} I_i^l C_i^d - \mu_k \bar{C_k^d} \right). \\ b_k &= \bar{C_k^d} - a_k^T \mu_k. \\ \widehat{C_i^d} &= \bar{a}_i^T I_i^l + \bar{b}_i. \end{aligned} \tag{4}$$

Here, $\bar{C_k^d} = \frac{1}{|\omega|} \sum_{i\in\omega_k} C_i^d$ is the mean of C^d in ω_k and $\bar{a}_i = \frac{1}{|\omega|} \sum_{i\in\omega_k} a_k$ and $\bar{b}_i = \frac{1}{|\omega|} \sum_{i\in\omega_k} b_k$. All summations are 3D box filters and can be computed in $O(N)$ time, where N is the number of image voxels.

2.3 Disparity Selection

After smoothing the cost volumes with the spatio-temporal guided filter, a spatio-temporal disparity volume f^l, which holds the sequence of disparity maps for the left input sequence I^l, is computed by applying the Winner-Takes-All strategy:

$$f_i^l = \operatorname*{argmin}_{d\in\mathcal{D}} \widehat{C_i^d}, \tag{5}$$

where $\mathcal{D}$ is the set of allowed disparities.

In addition to the disparity volume for the left video volume I^l, we also compute the disparity volume f^r for the right input video volume I^r in a similar manner by substituting I^l with I^r and vice versa in eqs. (1)-(4). Then we apply left-right consistency checking: A pixel in the left disparity volume f^l is marked as invalid, if the disparity value of its matching pixel in f^r differs by a value > 1 pixel. The invalid pixels are then filled by the lowest disparity value of the closest valid pixels which lie on the same spatial scanline (i.e. a single row in x dimension). This simple filling can generate artifacts in the output disparity maps. To reduce these artifacts, we apply a spatio-temporal weighted median filter on the *filled* regions. This weighted median filtering smoothes the filled pixels, while preserving the object boundaries. In particular, we use a filter kernel with dimensions $w_x^b \times w_y^b \times w_t^b$ with bilateral filter weights:

$$W_{i,j}^{BL} = \frac{1}{K_i} \exp(-\frac{|i-j|^2}{\sigma_s^2}) \exp(-\frac{|I_i - I_j|^2}{\sigma_c^2}), \tag{6}$$

where K_i is a normalization factor and σ_s, σ_c are parameters which adjust the spatial and color dissimilarity, respectively.

3 Temporally Consistent Optical Flow

We now adopt our temporal stereo matching framework to optical flow estimation. Similar to stereo, the goal is to reduce the ambiguity of the solution space by exploiting the temporal coherence of the flow field. The implicit assumption of our algorithm is that the flow vectors are constant within self-similar regions of the video volume. This assumption is usually met for objects that do not quickly change their speed and direction.

Our optical flow algorithm is almost identical to our stereo method. Here, the displacement vector u, used in the terms M^c and M^g in eq. (1), is defined as $u = (a, b, 0)$, where a and b is the flow in x and y directions, respectively. For constructing the cost volume we modify the stereo correlation measure, by additionally using the gradient in y direction. This is done by replacing term M^g in eq. (1) with

$$M_{i,u}^g = \min\left(\|\nabla_x I_i^l - \nabla_x I_{i-u}^r\| + \|\nabla_y I_i^l - \nabla_y I_{i-u}^r\|, \tau_g\right). \tag{7}$$

Once the cost volumes are established, we filter them and obtain the flow field by applying the Winner-Takes-All strategy as in sec. 2.3. As for stereo, we apply left-right consistency checking to determine invalid matches[1]. We then follow [1] and fill the invalid matches by applying a 2D (spatial) weighted median filter of size $w_x^b \times w_y^b$ with weights as in eq. (6). Since the weighted median filter overlaps some valid pixels, the flow vectors can be propagated to the invalid regions. To compute sub-pixel accurate flow fields, we simply upscale the input frames using bicubic interpolation as done in [26],[1].

4 Experimental Results

We implemented and tested our proposed temporal stereo and optical flow algorithms on a PC equipped with an Intel Core 2 Quad, 2.4 GHZ CPU and an Nvidia GeForce GTX480 GPU with 1.5GB of memory. We used CUDA for implementing our algorithm on the GPU.

In our test runs, the parameters of our algorithm were set to constant values. We use the following constant parameter settings to generate all results for both stereo matching and optical flow: $\{w_x{=}w_y,\ w_t{=}w_t^b,\ w_x^b{=}w_y^b,\ \epsilon,\ \alpha,\ \sigma_s,\ \sigma_c,\ \tau_c,\ \tau_g\}$ $= \{31, 5, 15, 0.001, 0.5, 9, 0.1, 0.028, 0.008\}$. These parameters have been found empirically.

[1] In optical flow literature it is called forward-backward consistency check.

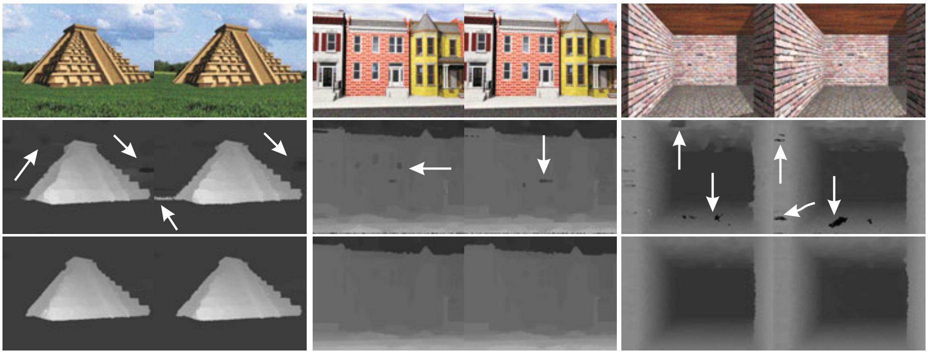


Fig. 3. Temporal vs. frame-by-frame processing. 1^{st} row: Left input frames of stereo sequences are shown. 2^{nd} row: Disparity maps computed by a frame-by-frame implementation show flickering artifacts (arrows point to major artifacts). 3^{rd} row: Our proposed method exploits temporal information, thus can remove most artifacts. Test sequences courtesy of [6].

Table 1. Quantitative stereo comparison. Our method outperforms all competitors in terms of quality and is the fastest method that maintains temporal coherent results (time measured without post-processing). See text for details.

Algorithm	Time[ms]	Book		Street		Tanks		Temple		Tunnel	
		mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev
DCBGrid [6]	51	52.2	2.1	32.5	2.3	36.0	6.2	39.5	1.9	25.7	11.1
Ours frame-by-frame	**41**	**18.0**	**1.5**	**16.4**	**1.3**	**10.8**	**1.7**	**11.17**	**2.1**	**14.4**	**7.8**
Temporal DCBGrid [6]	90	44.0	2.0	25.9	2.0	31.4	6.1	31.7	1.8	36.4	7.9
Ours Temporal	**41**	**10.1**	**1.5**	**12.2**	**1.0**	**8.7**	**1.3**	**6.6**	**1.5**	**17.7**	**8.6**

4.1 Stereo

We evaluated our stereo method visually on real stereo video sequences as well as quantitatively using a synthetic dataset comprising five stereo sequences with known ground truth disparity, provided by [6]. We compare our spatio-temporal stereo method to both the frame-by-frame and the spatio-temporal method of [6], which we denote as "DCBGrid" and "Temporal DCBGrid", respectively (we used the authors implementation to generate results for [6]). A further competitor is our approach applied in a frame-by-frame manner, i.e. using a temporal filter window of size 1 ($w_t = w_t^b = 1$ in eq. (3) and eq. (6)). Note that our method applied on a frame-by-frame basis degenerates to the method of [1].

First, we visually compare the results of our spatio-temporal method with the frame-by-frame variant of our approach. Fig. 3 shows results for two consecutive frames of three synthetic sequences. The disparity maps generated by our spatio-temporal method (3^{rd} row in fig. 3) are temporally coherent and exhibit less artifacts than the disparity maps generated by our method applied in a frame-by-frame manner (2^{nd} row in fig. 3).

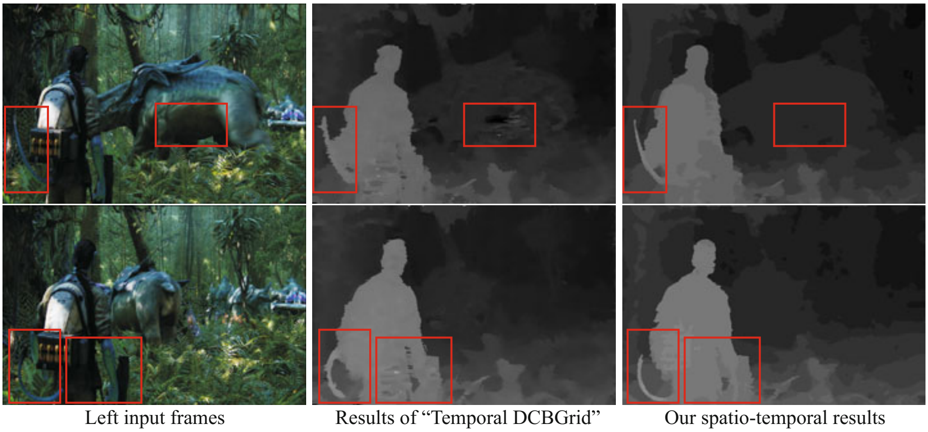


Fig. 4. Visual comparison to "Temporal DCBGrid" [6] Example (1). On these two frames of a movie sequence our temporal regularization could remove artifacts better than the temporal approach of [6] (red boxes mark major differences).

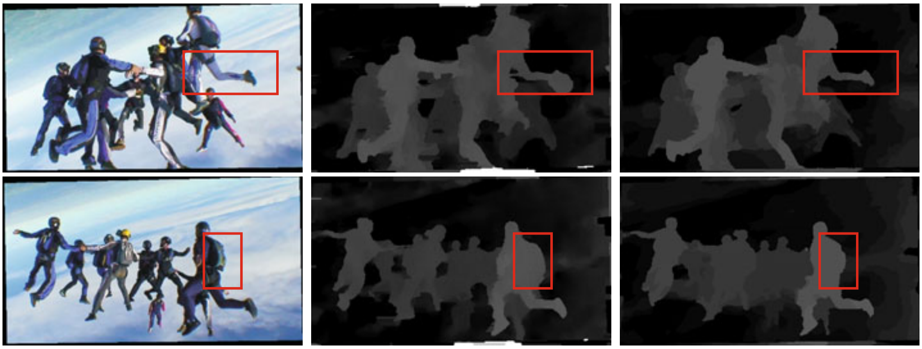
Next, we quantitatively compare our approach to its competitors. To this end, we follow [6] and generate results for the dataset of [6] after adding Gaussian noise[2] of ($\sigma = 20$) to the input frames. Table 1 shows the mean error (percentage of bad pixels) for the five ground truth sequences as well as their standard deviations. The table shows that our spatio-temporal method outperforms the "Temporal DCBGrid" in terms of mean error and smaller standard deviations[3]. This indicates that our algorithm gives results of higher quality and is more robust. This is also reflected in the visual comparison shown in figs. 4 and 5. Results for the full video sequences are shown in the supplementary material. Table 1 also reveals that our temporal method is more than two times faster[4] than the "Temporal DCBGrid". An important note from this table is that the time consumed by our frame-by-frame method is the same as the time of our temporal method. This is mainly due to the fact that in the 3D box filter the sliding window technique is used in the three dimensions w_x, w_y and w_t.

In another experiment, we evaluate the robustness of the different methods to noise. In particular, we plot the error rates at different noise levels (additive Gaussian noise with σ in the range from 0 to 100) for each of the five sequences of [6] in fig. 6. We see that for a noise level of zero, all methods are close-by. This is because the synthetic scenes are absolutely noise-free (which is in contrast to real-world scenarios), which makes it relatively easy for the different

[2] This was done in order to provide a more realistic scenario, since the original images are noise-free.

[3] The two algorithms have poor performance in the "Tunnel" sequence. A possible explanation is that this is due to that this sequence is highly textured and the filtering process over-smoothes its frames.

[4] Times were measured on different machines but should give a good indication of the computational complexity.



Left input frames Results of "Temporal DCBGrid" Our spatio-temporal results

Fig. 5. Visual comparison to "Temporal DCBGrid" [6] Example (2). Our spatio-temporal method can preserve some object outlines better than the approach of [6].

approaches to perform well. However, as the noise level increases the error level for the "DCBGrid" and the "Temporal DCBGrid" increase faster than the error level for our methods. Furthermore, the "Temporal DCBGrid" only slightly improves over its frame-by-frame counterpart ("DCBGrid"). In contrast, the improvement gained by our temporally coherent algorithm over the frame-by-frame implementation of our approach is considerably larger.

4.2 Optical Flow

We follow previous work [24] and test our approach on the "Yosemite" and "Marble" sequences which can be obtained from http://www.cs.brown.edu/ black/ images.html and http://i21www.ira.uka.de/image_sequences/, respectively. We did not test on the Middlebury sequences, because they violate our assumption of a temporally smooth flow field. (Note that also previous work did not test on these sequences for the same reason [24].)

Table 2. Quantitative flow comparison. The average angular error (AAE) for different temporal window sizes on the "Yosemite" and "Marble" sequences is shown.

Temporal Window Size (w_t)	1	3	5	7
Yosemite Mean AAE	3.85°	2.90°	2.83°	2.76°
Marble Mean AAE	6.72°	5.26°	4.78°	4.61°

We show the Average Angular Error (AAE) for different sizes of the temporal filter window w_t on the "Yosemite" and "Marble" sequences in table 2. We see that an increased temporal window size notably reduces the error. (Note that when using a temporal filter window of size 1 our method degenerates to the approach in [1].) Hence, our temporal processing improves the results of [1]. We

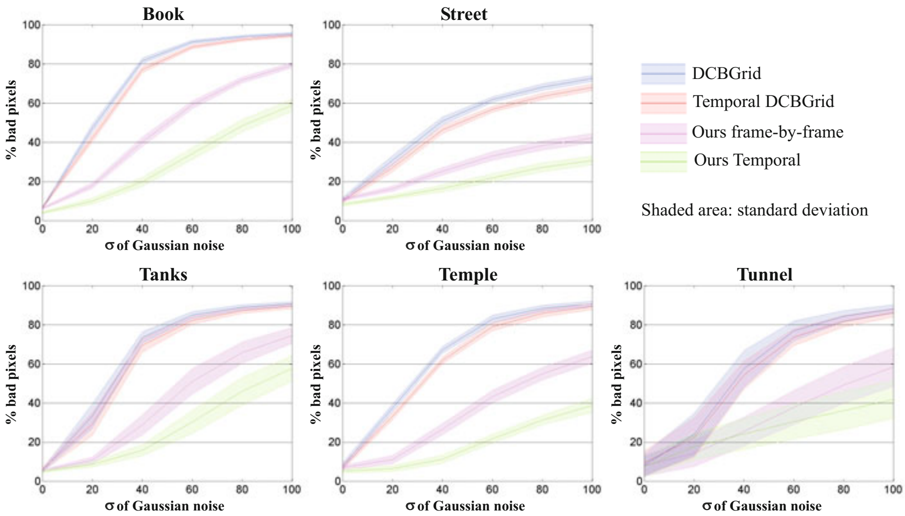
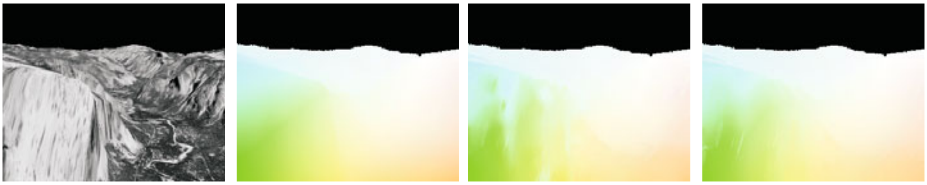


Fig. 6. Robustness to noise. We plot the average error vs. Gaussian noise levels for five ground truth sequences. See the text for a discussion and the supplementary material for a visual comparison. Best viewed in color.



(a) Input frame (b) Ground truth (c) Our frame-by-frame result (d) Our temporal result

Fig. 7. Effect of temporal smoothing for optical flow. Temporally smoothing the flow field gives a result that is closer to the ground truth than the flow generated without temporal coherency. Color coding of flow maps as on the Middlebury evaluation page.

visually show the effect of the temporal processing for frame 8 of the "Yosemite" sequence in fig. 7. The optical flow map generated with our spatio-temporal method is closer to the ground truth than the one computed by our frame-by-frame method.

5 Conclusion

This paper proposed an efficient technique for computing temporally coherent disparity maps from a sequence of stereo images. The main contribution was to extend the filter-based frame-by-frame stereo approach of [1] to the temporal domain. To this end we adopted the 2D image filter of [5] to the 3D spatio-temporal space. A further contribution was to apply our method to optical flow estimation. We demonstrated that exploiting temporal information considerably improves the frame-by-frame approach of [1] for both stereo and optical flow estimation and we outperform the current state-of-the-art in local space-time stereo matching.

Future work may concentrate on obtaining a better understanding of the relationship between filtering-based and energy-based optimization. This knowledge may lead to fast and even better filtering approaches than presented in this paper.

References

1. Rhemann, C., Hosni, A., Bleyer, M., Rother, C., Gelautz, M.: Fast Cost-Volume Filtering for Visual Correspondence and Beyond. In: CVPR (2011)
2. Yoon, K.J., Kweon, I.S.: Locally Adaptive Support-Weight Approach for Visual Correspondence Search. In: CVPR (2005)
3. Hosni, A., Bleyer, M., Gelautz, M., Rhemann, C.: Local Stereo Matching Using Geodesic Support Weights. In: ICIP (2009)
4. Hosni, A., Bleyer, M., Gelautz, M.: Near Real-Time Stereo With Adaptive Support Weight Approaches. In: 3DPVT (2010)
5. He, K., Sun, J., Tang, X.: Guided Image Filtering. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 1–14. Springer, Heidelberg (2010)
6. Richardt, C., Orr, D., Davies, I., Criminisi, A., Dodgson, N.A.: Real-time Spatiotemporal Stereo Matching Using the Dual-Cross-Bilateral Grid. In: Daniilidis, K. (ed.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 510–523. Springer, Heidelberg (2010)
7. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV 47(1/2/3), 7–42 (2002), http://www.middlebury.edu/stereo/
8. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-Dimensional Scene Flow. In: ICCV (1999)
9. Leung, C., Appleton, B., Lovell, B.C., Sun, C.: An Energy Minimisation Approach to Stereo-Temporal Dense Reconstruction. In: ICPR (2004)
10. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: CVPR (2010)
11. Werlberger, M., Pock, T., Bischof, H.: Motion estimation with non-local total variation regularization. In: CVPR (2010)
12. Salgado, A., Sánchez, J.: Temporal Constraints in Large Optical Flow Estimation. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2007. LNCS, vol. 4739, pp. 709–716. Springer, Heidelberg (2007)
13. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
14. Weickert, J., Schnörr, C.: Variational optic flow computation with a spatio-temporal smoothness constraint. In: JMIV, vol. 14, pp. 245–255 (2001)
15. Black, M.J., Anandan, P.: Robust dynamic motion estimation over time. In: CVPR (1991)

16. Nagel, H.H.: Extending the 'Oriented Smoothness Constraint' into the Temporal Domain and the Estimation of Derivatives of Optical Flow. In: Faugeras, O. (ed.) ECCV 1990. LNCS, vol. 427, pp. 139–148. Springer, Heidelberg (1990)
17. Zhang, L., Curless, B., Seitz, S.M.: Spacetime Stereo: Shape Recovery for Dynamic Scenes. In: CVPR (2003)
18. Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: high-resolution capture for modeling and animation. In: SIGGRAPH (2004)
19. Davis, J., Nehab, D., Ramamoorthi, R., Rusinkiewicz, S.: Spacetime stereo: a unifying framework for depth from triangulation. In: PAMI, vol. 27(2), pp. 296–302 (2005)
20. Jenkin, M., Tsotsos, J.: Applying temporal constraints to the dynamic stereo problem. In: CVGIP, vol. 33, pp. 16–32 (1986)
21. Williams, O., Isard, M., MacCormick, J.: Estimating Disparity and Occlusions in Stereo Video Sequences. In: CVPR (2005)
22. Larsen, E.S., Mordohai, P., Pollefeys, M., Fuchs, H.: Temporally consistent reconstruction from multiple video streams using enhanced belief propagation. In: ICCV (2007)
23. Bleyer, M., Gelatuz, M.: Temporally Consistent Disparity Maps from Uncalibrated Stereo Videos. In: ISPA (2009)
24. Zimmer, H., Bruhn, A., Weickert, J.: Optic Flow in Harmony. In: IJCV, vol. 93, pp. 368 – 388 (2011)
25. Sizintsev, M., Wildes, R.P.: Spatiotemporal stereo via spatiotemporal quadric element (stequel) matching. In: CVPR (2009)
26. Steinbrücker, F., Pock, T., Cremers, D.: Large displacement optical flow computation without warping. In: ICCV (2009)

Specular-Free Residual Minimization for Photometric Stereo with Unknown Light Sources

Tsuyoshi Migita, Kazuhiro Sogawa, and Takeshi Shakunaga

Okayama University

Abstract. We address a photometric stereo problem that has unknown lighting conditions. To estimate the shape, reflection properties, and lighting conditions, we employ a nonlinear minimization that searches for parameters that can synthesize images that best fit the input images. A similar approach has been reported previously, but it suffers from slow convergence due to specular reflection parameters. In this paper, we introduce specular-free residual minimization that avoids the negative effects of specular reflection components by projecting the residual onto the complementary space of the light color. The minimization process simultaneously searches for the optimal light color and other parameters. We demonstrate the effectiveness of the proposed method using several real and synthetic image sets.

1 Introduction

Photometric stereo is a method for recovering the shape and albedo of an object from a set of images, when the object and the camera are fixed but the lighting conditions vary between images. A classical formulation assumes that the object is a Lambertian surface and that the lighting conditions are known. Several recently developed methods consider unknown lighting conditions and/or non-Lambertian surfaces (e.g., [4,14,3,10,2,11,12]).

We have developed a method based on the formulation of Migita et al. [6], that directly minimizes a cost function to estimate the shape and the reflection properties of the object and the light position for each image. The cost function is the difference between input and synthesized images based on the Torrance-Sparrow model [13]. In this study, we modify the cost function to improve the performance because the nonlinearity of the Torrance-Sparrow model causes the cost function to be highly nonlinear and thus convergence tends to be slow.

This method involves decomposing the residual (an RGB vector of the difference for each pixel) into two subspaces: a 1d space parallel to the light color and a 2d space orthogonal to the light color. This decomposition is similar to that proposed by Zickler et al. [15]. By using the latter component only, the specular term is removed so that the estimation should be faster and more accurate. In addition, removing specular component reduces the number of parameters to be estimated.

However, there are some disadvantages with removing nonlinear components from the input images. For example, the method cannot be used for monochrome

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 178–189, 2011.

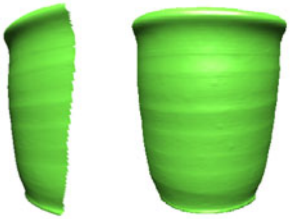


Fig. 1. Reconstructed shape

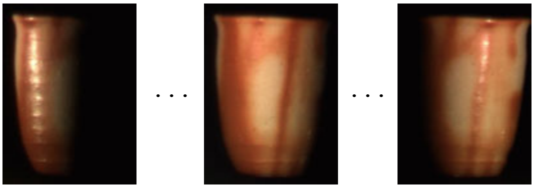


Fig. 2. Images under various lighting conditions

images. Another problem is the generalized bas-relief (GBR) ambiguity [1,7,11]. Since our method removes the specular component, which may help resolve the ambiguity, we have to rely only on the effect of near light sources to resolve the ambiguity.

The new formulation does not use the Torrance-Sparrow model. Instead the dichromatic reflection model [9,5] is used. This model is commonly used to analyze a color histogram to estimate the light source color. However, in our method, estimation of the light source color is integrated with the minimization process so that it is not a separate process.

We verify the effectiveness of our method using several real and synthetic image sets.

2 Formulation

This section describes our formulation, which is referred to as the specular-free residual (SR) minimization hereafter. We describe the similarities and differences between our method and another method [6], which we call the full-color residual (FR) minimization.

The method is essentially a photometric stereo method, which recovers the object shape (e.g., Fig. 1) from images obtained under various lighting conditions, such as Fig. 2. Input images are of a static object and are obtained by a static camera. We also assume that the lighting is a single point light source in the near distance. We need to estimate the shape and the reflection properties of the object and the light position for each image. The reconstruction is performed by nonlinear minimization by comparing input with images synthesized using an image generation model (see below).

2.1 Full-Color Residual Minimization

Each pixel in the input image corresponds to a surface element of the object, and its intensity is described by a certain reflection model.

The FR minimization uses a simplified version of the Torrance-Sparrow model [13] to describe the input intensity $\boldsymbol{e}_{fp}$ (an RGB vector for the p'th pixel in the f'th image) as follows:

$$\boldsymbol{e}_{fp} = \eta_f \boldsymbol{d}_p \cos\beta_{fp} + \eta_f s_p \boldsymbol{S} \frac{1}{\cos\gamma_p} \exp\left(\rho\alpha_{fp}^2\right) + \boldsymbol{r}_{fp} \tag{1}$$

where

$$\begin{aligned} \cos\beta_{fp} &= \mathcal{N}\left[\boldsymbol{n}_p\right]^T \mathcal{N}\left[\boldsymbol{l}'_{fp}\right], & \boldsymbol{l}'_{fp} &= \boldsymbol{l}_f - \boldsymbol{x}_p, \\ \cos\gamma_p &= \mathcal{N}\left[\boldsymbol{n}_p\right]^T \mathcal{N}\left[\boldsymbol{v}'_p\right], & \boldsymbol{v}'_p &= (\boldsymbol{v} - \boldsymbol{x}_p)/\ell, \\ \cos\alpha_{fp} &= \mathcal{N}\left[\boldsymbol{n}_p\right]^T \mathcal{N}\left[\mathcal{N}\left[\boldsymbol{l}'_{fp}\right] + \mathcal{N}\left[\boldsymbol{v}'_p\right]\right], & \eta_f &= \left|\boldsymbol{l}_f - \boldsymbol{x}_p\right|^{-2}, \end{aligned} \tag{2}$$

and $\boldsymbol{r}_{fp}$ is the residual, $\boldsymbol{n}_p$ is the normal vector, $\boldsymbol{l}_f$ is the light position, $\boldsymbol{v}$ is the camera position, ℓ is the focal length of the camera, $\boldsymbol{d}_p$ is an RGB vector describing the diffuse reflectance, s_p is the specular reflectance, $\boldsymbol{S}$ is an RGB vector describing the light color, η_f is a coefficient describing the attenuation of the light intensity due to the distance between the light and the object, ρ is a specular parameter, and $\mathcal{N}[\cdot]$ is an operator that normalizes the norm of a vector to 1.

Note that for the first term in eq. (1), $|\beta_{fp}| \geq \pi/2$ implies that the surface element is in an attached shadow region, which means the light is not positioned in front of the surface element. Consequently, this term is replaced with 0 in this case.

In the FR minimization, the shape, the reflection properties and the lighting conditions are reconstructed by minimizing the following cost function.

$$E(\boldsymbol{p}) = \frac{1}{2} \sum_{f=1}^{F} \sum_{p=1}^{P} |\boldsymbol{r}_{fp}|^2 \tag{3}$$

where $\boldsymbol{r}_{fp}$ is the residual term in eq. (1). Note that, when the surface element for the p'th pixel in the f'th image is judged to be saturated or too dark (may be due to a cast shadow), we set $\boldsymbol{r}_{fp} = (0,0,0)^T$. The minimization is performed by Levenberg-Marquardt (LM) method [8], the details of which are given in section 3.

2.2 Specular-Free Residual Minimization

In the FR minimization, the specular term (i.e. the second term in eq. (1)) strongly reduces the convergence rate of the LM minimization process. Furthermore, it is computationally very expensive to calculate its derivative function, which is required for the minimization. Thus, the basic idea of the present study is to remove the specular term by using the following cost function instead of eq. (3):

$$E(\boldsymbol{p}) = \frac{1}{2} \sum_{f=1}^{F} \sum_{p=1}^{P} \left| \frac{\boldsymbol{S}_\times}{|\boldsymbol{S}|} \boldsymbol{r}_{fp} \right|^2 \tag{4}$$

where

$$\boldsymbol{S}_\times = \begin{bmatrix} 0 & -S_3 & S_2 \\ S_3 & 0 & -S_1 \\ -S_2 & S_1 & 0 \end{bmatrix} \tag{5}$$

with $\boldsymbol{S} = (S_1, S_2, S_3)^T$. Since $\boldsymbol{S}_\times \boldsymbol{S} = 0$, we can calculate the specular-free residual $\boldsymbol{S}_\times \boldsymbol{r}_{fp}$ without calculating the second term in eq. (1). Thus, in our method, specular reflection is not limited to the Torrance-Sparrow model. Instead, the method employs a dichromatic reflection model; i.e., the specular color is same in every pixel, although its scale can differ between pixels.

However, this formulation has several drawbacks, because the cross product operation removes a part of the diffuse component in addition to the specular term. Consequently, some important information is lost. For example, it cannot process monochrome input images. Full-color input images are required to decompose the residual vectors into components parallel and orthogonal to the light color. Specifically the object must have two or more colors besides the light color to avoid a local minima that causes the estimator for $\boldsymbol{S}$ to converge to the diffuse color rather than the specular color.

In addition, the estimation is greatly stabilized by normalizing the residual terms in eq. (4) as follows:

$$E(\boldsymbol{p}) = \frac{1}{2} \sum_{f=1}^{F} \sum_{p=1}^{P} \frac{|\boldsymbol{S}_\times \boldsymbol{r}_{fp}|^2}{|\boldsymbol{S}_\times \boldsymbol{y}_p|^2} \ , \quad \text{where} \quad \boldsymbol{y}_p = \mathcal{N}\left[\sum_f \boldsymbol{e}_{fp}\right] \ . \tag{6}$$

This normalization is interpreted as follows. Applying $\boldsymbol{S}_\times$ from left, the input intensity is somewhat scaled and the scale factor is dependent on the color of the corresponding surface element. This scale factor should be compensated to accurately evaluate the residual $\boldsymbol{r}_{fp}$.

Another problem that needs to be considered is the GBR ambiguity. This ambiguity is described as follows. The Lambertian component is expressed in a bilinear form $\boldsymbol{l}^T\boldsymbol{n}$, which could be transformed into $(\boldsymbol{A}^{-T}\boldsymbol{l})^T(\boldsymbol{A}\boldsymbol{n})$ by any nonsingular 3×3 matrix $\boldsymbol{A}$. Using the integrability constraint, $\boldsymbol{A}$ can be specified up to three degrees of freedom [1]. To determine the remaining three parameters, previous studies have used several nonlinearities [7,2,10]. The FR minimization has some such nonlinearities. However, since our SR minimization removes the specular component, we have to use another clue to resolve the ambiguity. The most important clue is the attenuation with distance between the light source and the object, as shown in [7]. Below, we present an experimental result that demonstrates that this nonlinearity can resolve the ambiguity.

2.3 Estimation Parameters

The parameters to be estimated form a large vector $\boldsymbol{p}$. The $\boldsymbol{p}$ consists of four kinds of parameters: the object shape, the reflection properties, the global parameters, and the light positions as

$$\boldsymbol{p} = \left(\boldsymbol{p}_w^T, \boldsymbol{p}_s^T, \boldsymbol{p}_m^T, \boldsymbol{p}_l^T\right)^T \ . \tag{7}$$

Object Shape. The object shape is described by the depth λ_p for each pixel. A vector containing all the depths is $\boldsymbol{p}_s = (\lambda_1, \cdots, \lambda_P)^T$, where P is the number of pixels to be estimated.

From the depths, the 3d coordinates $\boldsymbol{x}_p$ for the p'th surface element are calculated using the following formula:

$$\boldsymbol{x}_p = \lambda_p(\frac{u_p}{\ell}, \frac{v_p}{\ell}, 1)^T + (u_p, v_p, 0)^T \tag{8}$$

where $(u_p, v_p)^T$ is the 2d coordinates of the pixel with respect to the image center and ℓ is the focal length of the camera. For affine camera model, ℓ is set to infinity, while for projective camera model, ℓ is set to a finite value. The world origin is located at the center of the image plane and the camera position $\boldsymbol{v}$ is $(0, 0, -\ell)^T$.

The surface normal $\boldsymbol{n}_p$ for the p'th pixel is calculated from the 3d coordinates of its neighboring pixels, l, r, t, b (i.e., left, right, top and bottom, respectively), as follows:

$$\boldsymbol{n}_p = (\boldsymbol{x}_t - \boldsymbol{x}_b) \times (\boldsymbol{x}_r - \boldsymbol{x}_l) \ . \tag{9}$$

The normal is not limited to being a unit vector. Instead, normalization is included in the cosine operations in eq. (2).

Reflectance. In our SR minimization, reflection parameters are the diffuse colors $\boldsymbol{d}_p$ for each pixel. Unlike the FR minimization, specular reflectance s_p is not required. Therefore, $\boldsymbol{p}_w = (\boldsymbol{d}_1^T, \cdots, \boldsymbol{d}_P^T)^T$.

Global Parameters. The only global parameter in the SR minimization is the light (or specular) color $\boldsymbol{S}$, whereas the FR minimization has an additional specular parameter, ρ.

Light Positions. The light position is estimated for each image. Thus, $\boldsymbol{p}_l = (\boldsymbol{l}_1^T, \cdots, \boldsymbol{l}_F^T)^T$, where F is the number of input images.

3 Minimization

Letting the number of pixels to be P, there are more than $4P$ elements in $\boldsymbol{p}$. Thus, the search space can typically have the dimension of about 100,000. An algorithm is required that can deal with a minimization problem on this scale.

3.1 LM Method

We use LM method for minimizing the cost function, as in the FR minimization, which is given by:

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k - (J_k^T J_k + \mu_k I)^{-1} J_k^T \boldsymbol{r}_k \tag{10}$$

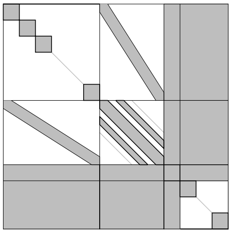
where $\boldsymbol{r}_k$ is a vector containing all the (specular-free) residual vectors, and J is the Jacobian matrix. The subscript k indicates the value is dependent on the parameter $\boldsymbol{p}_k$. The initial value $\boldsymbol{p}_0$ is discussed in section 3.3.

Each (specular-free) residual depends on only 14 parameters: the diffuse reflectance, the depths of neighboring pixels, the light color, and the light position. Thus, the Jacobian contains 14 non-zero elements in each row. The following equation for the total derivative of $\boldsymbol{r}_{fp}$ contains all the non-zero entries in J_k.

$$\delta \boldsymbol{r}_{fp} = \frac{\partial \boldsymbol{r}_{fp}}{\partial \boldsymbol{d}_p} \delta \boldsymbol{d}_p + \frac{\partial \boldsymbol{r}_{fp}}{\partial \lambda_p} \delta \lambda_p + \cdots + \frac{\partial \boldsymbol{r}_{fp}}{\partial \boldsymbol{l}_f} \delta \boldsymbol{l}_f \ . \tag{11}$$

The Hessian matrix can then be easily calculated. Its structure is given by:

$$J_k^T J_k = \tag{12}$$



Each block contains (from left to right or top to bottom) approximations of the second order derivatives corresponding to $\boldsymbol{p}_w$, $\boldsymbol{p}_s$, $\boldsymbol{p}_m$, and $\boldsymbol{p}_l$, respectively.

For each iteration, we have to solve a linear system with this coefficient matrix. Although this matrix is large, it is relatively sparse. To exploit its sparse structure, it is preferable to use a preconditioned conjugate gradient method [8] to solve the system.

3.2 Preconditioned Conjugate Gradient Method

We use the following preconditioned conjugate gradient method to solve $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$:

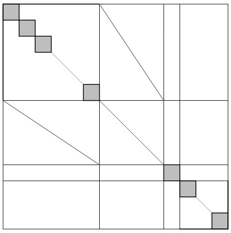
$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \boldsymbol{d}_k \tag{13}$$

where

$$\begin{aligned}
\boldsymbol{d}_k &= \begin{cases} \boldsymbol{C}^{-1}\boldsymbol{g}_0 & (k=0) \\ \boldsymbol{C}^{-1}\boldsymbol{g}_k + \beta_k \boldsymbol{d}_{k-1} & (k>0) \end{cases} \\
\boldsymbol{g}_k &= \boldsymbol{A}\boldsymbol{w}_k - \boldsymbol{b} \\
\beta_k &= \boldsymbol{g}_k^T \boldsymbol{C}^{-1} \boldsymbol{g}_k / \boldsymbol{g}_{k-1}^T \boldsymbol{C}^{-1} \boldsymbol{g}_{k-1} \\
\alpha_k &= \boldsymbol{d}_k^T \boldsymbol{C}^{-1} \boldsymbol{g}_k / \boldsymbol{d}_k^T \boldsymbol{A} \boldsymbol{d}_k
\end{aligned}$$

Here, $\boldsymbol{C}$ is a preconditioner, which should be an approximation of $\boldsymbol{A}$ and computationally easy to invert. This is constructed by taking the block diagonal part of $\boldsymbol{A}$, or $J_k^T J_k + \mu_k I$ in eq. (10), as:

$$\boldsymbol{C} = \tag{14}$$



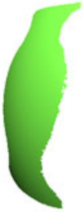


Fig. 3. Initial shape

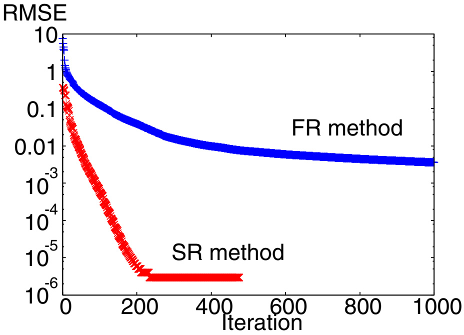


Fig. 4. Residual w.r.t. iteration

3.3 Initialization

The initial parameter of the LM process is as follows.

Object Shape. We initialize the shape as a simple paraboloid described by the following formula by an appropriate set of parameters (a, b):

$$\lambda_p = au_p^2 + bv_p^2 \tag{15}$$

Fig. 3 shows an example. Using an appropriate curvature is effective for avoiding local minima. Otherwise a convex shape may be recovered as a concave object.

Lighting. There are two types of lighting parameters: color, which is initialized to $(1, 1, 1)^T$ (i.e., a white light source), and the light position for each image, which is initialized to $(0, 0, -d)^T$, which is independent of f, where d is an appropriate value.

Although this is a very rough initialization, the nonlinear optimization algorithm can search for a reasonable shape and light positions. A more elaborate initialization such as a method based on SVD [14] may improve the convergence.

Reflectance. The Lambertian parameter $\boldsymbol{d}_p$ is an RGB vector that is a scalar multiple of the reflectance. It is initialized by taking an average of the input images as in the following formula.

$$\boldsymbol{d}_p = \frac{1}{F}\sum_{f=1}^{F} \boldsymbol{e}_{fp} \tag{16}$$

4 Experiments

The SR minimization was tested on several real and synthetic image sets.

In several of the experiments, affine and projective camera models were tested and the results obtained were similar. The results given below are based on the affine model (i.e., infinite ℓ). Note that the computational cost for infinite ℓ could be significantly smaller than for a finite ℓ, although this is not the focus of this paper.

4.1 Convergence Speed

The SR minimization removes the specular components to speed up convergence. Fig. 4 shows the convergence speeds for FR and SR minimizations for an synthetic image set. The SR minimization converges much faster than the FR minimization. Although comparison of absolute RMSE values are not so meaningful because the definitions of the residuals are different, the RMSE cannot be significantly lower than 10^{-6}, since the input images are given in 32-bit floating point format. The SR minimization reached this limit after about 200 iterations, but we doubt that the FR minimization will reach this limit.

4.2 Resistance for GBR Ambiguity

We verified that the nearby lighting can resolve GBR ambiguity in our formulation. To demonstrate this, we used two sets of synthetic images. One is rendered using near light sources and the other using distant light sources. Several minimization trials were conducted using various initial values that were created by applying different GBR transformations to the same converged parameters. Fig. 5 shows the results. The bottom row corresponds to the cases for near light sources. The same distinctive shape was obtained from several initial values. In contrast, the reconstructed shapes were not fully corrected for distant light sources (the top row).

4.3 Real Image Sets

Here, we present several experiments on real image sets. We stopped the search after 200 iterations. The results fully converged for several image sets, but not for other image sets.

In these experiments, we used multiple light sources that had different intensities, despite the fact that eq. (1) assumes that every light source has the same intensity. This might generate some error in the distance between the light source and the target object.

Apple. A fresh apple was imaged in a dark room. The light positions are shown in Fig. 6. We obtained 24 images, several examples of which are shown in Fig. 7. The size of each image is 125 × 133 pixels, and 11,098 points are estimated. The FR and SR minimizations produced the results shown in Fig. 8 and Fig. 9, respectively. In Fig. 8, the estimation produced a concave object, which is apparently a failure. This is due to an intense specular reflection near the image center. In Fig. 9, on the other hand, the estimated shape appears smooth and apple-like. This result is an example of the SR minimization outperforming the FR minimization.

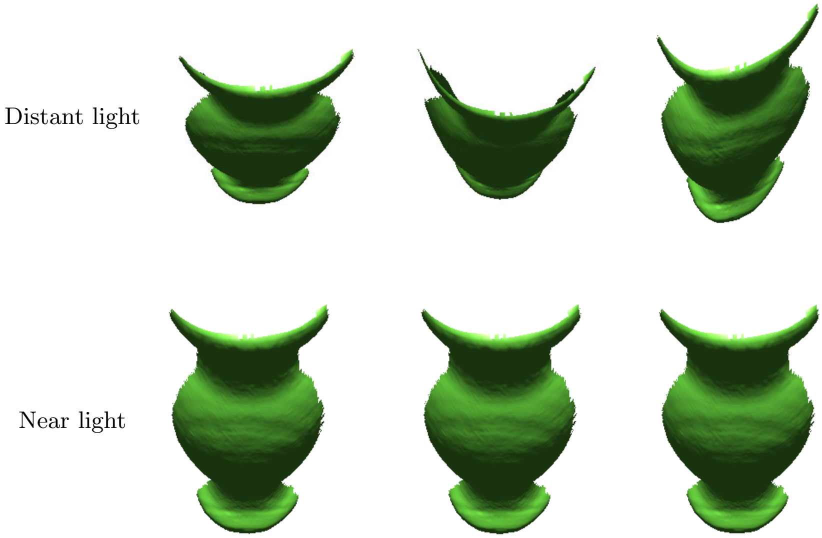


Fig. 5. Shapes reconstructed from several initial values

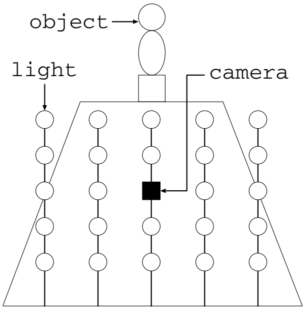


Fig. 6. Configuration

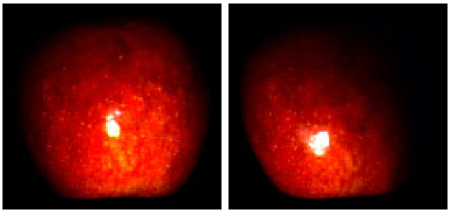


Fig. 7. Example images of an apple

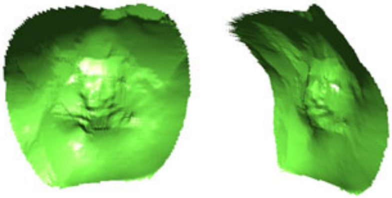


Fig. 8. Apple results by FR minimization

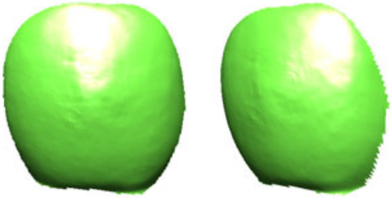


Fig. 9. Apple results by SR minimization

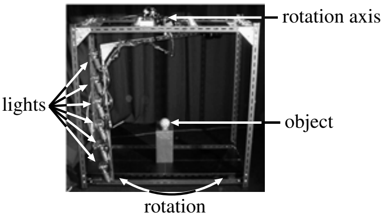


Fig. 10. Lighting system

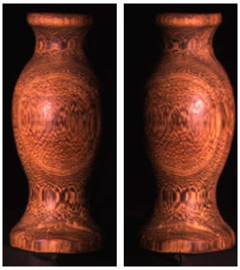


Fig. 11. Example images of a wooden figure

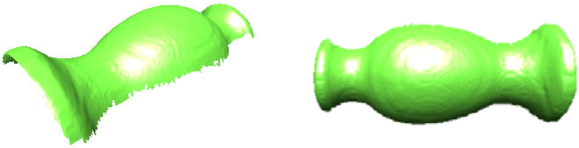
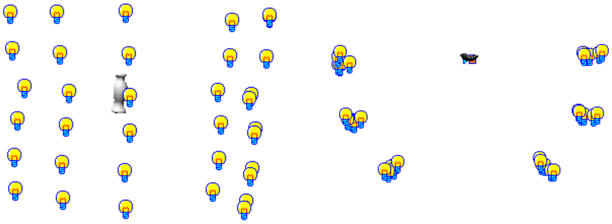


Fig. 12. Results for a wooden figure



(a) Side view (b) Top view

Fig. 13. Light positions and a wooden figure

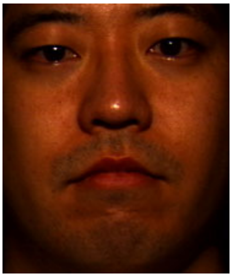


Fig. 14. An image of a human face

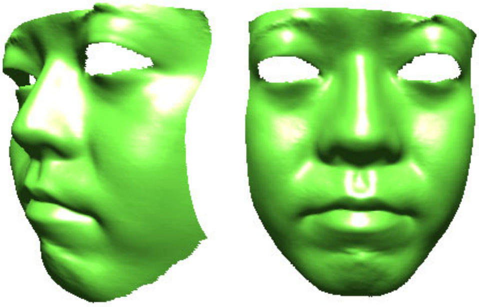


Fig. 15. Results for a human face

Wooden Figure. A wooden figure was imaged using the lighting system shown in Fig. 10, which has six light bulbs on a rotating arm. We obtained 36 images (Fig. 11) that were 128 × 296 in size and 25,480 pixels were used for the estimation. The resulting shape is shown in Fig. 12, and the light positions are shown in Fig. 13 along with the reconstructed object. We used this sequence because the light positions are known to form a cylinder. Fig. 13 shows slightly distorted cylinder. One possible cause for this distortion is that the light bulbs have different intensities. On the other hand, the angles between the light columns are estimated well, as the top view image shows. The radius and the height of the estimated cylinder divided by the object height are approximately 3.2–4.2 and 4.6–5.5, respectively. There are considered to be reasonable results compared to their actual values of 3.7 and 5.0, respectively.

Human Face. We show an example of applications to human faces. The input images, an example of which is shown in Fig. 14, were obtained in the environment shown in Fig. 6. There are 24 images with a size 185 × 220, and 35,629 pixels of which were used for the estimation. The eye regions were manually removed because they were too glossy. The result is shown in Fig. 15. It is a satisfactory result, since it does not exhibit any severe degradation due to specular reflection and/or shadows cast around the nose.

5 Conclusions

This study considered a photometric stereo problem with unknown lighting conditions. It has been reported [3,6] that the shape and the reflection properties of the object and the lighting conditions can be recovered by minimizing the differences between input and estimated images based on a reflection model. In this paper, the cost function of the minimization is not constructed from the full-color residual, but from a specular-free projection of the residual in a space orthogonal to the light color. Using several real and synthetic image sets, we demonstrated that the specular-free residual (SR) minimization exhibits better performance than the full-color residual (FR) minimization for several cases.

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant-in-Aid for Young Scientists (B) (No. 22700181).

References

1. Belhumeur, P.N., Kriegman, D.J., Yuille, A.L.: The bas-relief ambiguity. Intl. J. of Computer Vision 35(1), 33–44 (1999)
2. Chandraker, M., Kahl, F., Kriegman, D.: Reflections on the generalized bas-relief ambiguity. In: Proc. CVPR 2005, vol. I, pp. 788–795 (2005)
3. Georghiades, A.S.: Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo. In: Proc. ICCV 2003, vol. 2, pp. 816–823 (2003)

4. Ikeuchi, K.: Determining surface orientations of specular surfaces by using the photometric stereo method. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-3(6), 661–669 (1981)
5. Klinker, G., Shafer, S., Kanade, T.: The measurement of high-lights in color images. Intl. J. Computer Vision 2(1), 7–32 (1988)
6. Migita, T., Ogino, S., Shakunaga, T.: Direct Bundle Estimation for Recovery of Shape, Reflectance Property and Light Position. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 412–425. Springer, Heidelberg (2008)
7. Okabe, T., Sato, Y.: Does a nearby point light source resolve the ambiguity of shape recovery in uncalibrated photometric stereo? In: Proc. MIRU 2007. pp. 881–886 (2007) (in Japanese)
8. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes, 3rd edn. Cambridge University Press (2007)
9. Shafer, S.A.: Using color to separate reflection components. COLOR Research and Application 10(4), 210–218 (1985)
10. Tan, P., Mallick, S.P., Quan, L., Kriegman, D., Zickler, T.: Isotropy, reciprocity and the generalized bas-relief ambiguity. In: Proc. CVPR 2007, pp. 1–8 (2007)
11. Tan, P., Zickler, T.: A projective framework for radiometric image analysis. In: Proc. CVPR 2009, pp. 2977–2984 (2009)
12. Tan, P., Lin, S., Quan, L.: Resolution-Enhanced Photometric Stereo. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part III. LNCS, vol. 3953, pp. 58–71. Springer, Heidelberg (2006)
13. Torrance, K., Sparrow, E.: Theory for off-specular reflection from roughened surfaces. J. Opt. Soc. Am. 57, 1105–1114 (1967)
14. Yuille, A., Snow, D., Epstein, R., Belhumeur, P.: Determining generative models of objects under varying illumination: Shape and albedo from multiple mages using svd and integrability. Intl. J. of Computer Vision 35(3), 203–222 (1999)
15. Zickler, T., Mallick, S.P., Kriegman, D.J., Belhumeur, P.N.: Color subspaces as photometric invariants. Intl. J. of Computer Vision 79(1), 13–30 (2008)

Analysing False Positives and 3D Structure to Create Intelligent Thresholding and Weighting Functions for SIFT Features

Michael May, Martin Turner, and Tim Morris

The University of Manchester, UK
michael.may@student.manchester.ac.uk
{martin.turner,tim.morris}@manchester.ac.uk
http://www.cs.manchester.ac.uk
http://www.michael-may.co.uk

Abstract. This paper outlines image processes for object detection and feature match weighting utilising stereoscopic image pairs, the Scale Invariant Feature Transform (SIFT) [13,4] and 3D reconstruction. The process is called FEWER; Feature Extraction and Weighting for Enhanced Recognition. The object detection technique is based on noise subtraction utilising the false positive matches from random features. The feature weighting process utilises a 3D spatial information generated from the stereoscopic pairs and 3D feature clusters. The features are divided into three different types, matched from the target to the scene and weighted based on their 3D data and spatial cluster properties. The weightings are computed by analysing a large number of false positive matches and this gives an estimation of the probability that a feature is matched correctly. The techniques described provide increased accuracy, reduces the occurrence of false positives and can create a reduced set of highly relevant features.

1 Introduction

The scale invariant feature transform (SIFT) [13] is used as a detection algorithm for finding correspondence between features within parts of images thereby allowing image matching to occur. In this paper we consider the specific matching problem of a target stereoscopic image pair of a 3D object within a hand-held stereoscopic video sequence. This paper introduces novel techniques for object detection and feature weighting. The process is called FEWER; Feature Extraction and Weighting for Enhanced Recognition.

For the detection process a set of random features are matched to the scene and the ratio of matches to the number of target features is used as a baseline for noise as these are false positives. Subtracting this noise correspondence ratio from the correspondence ratio calculated from a target image acts as a threshold to indicate if the object is present in a scene.

For the weighting process a 3D point cloud is constructed from target and scene stereo pairs and the features are clustered. For each image the features are divided into three different types, matched from the target to the scene and

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 190–201, 2011.

weighted based on the their 3D and spatial cluster properties. This weighting gives an estimation of the probability that a feature is matched correctly. The technique, similar to the previous one, utilises the expected rate of false positives found by studying how randomly selected features match to a scene, creating noise property statistics.

The paper is structured as follows; background work, an explanation of the noise subtraction based object detection, the feature weighting process, with an explanation of the technique by which the weightings are calculated, followed by evaluation and conclusions.

1.1 Background

The SIFT feature detection algorithm developed and pioneered by David Lowe [4,13] is a process that creates unique and highly descriptive features from an image. These features are designed to be invariant to rotation and are robust to changes in scale, illumination, noise and small changes in viewpoint. The features are used to indicate if there is any correspondence between areas within images. This allows object recognition to be implemented by comparing a set of features generated from input images to a set of features generated from images of target objects.

As the target and scene data both consist of stereoscopic pairs a structure from motion (SfM) system (Bundler API utilising a modified version of the sparse bundle adjustment [7] as the optimisation engine) is used to detect different types of matches and produce 3D geometric reconstruction.

Object recognition work using multiple views of a scene has been carried out [5,8,18] using multiple images and rough registration information to determine possible corresponding detections across multiple viewpoints. Work on integrating information across many images has been conducted using Bayesian strategies to combine uncertain information between views [10,19]. Combining data across multiple frames of a video to obtain depth information has also been studied [1,20]. Many other papers show that the use of 3D depth information [3,6,8,11,12,16,17] can be applied successfully to aid object recognition.

Although the processes in this paper use SIFT they could be applied to many other feature detectors such as SURF [2], GLOH [15] or FAST [9].

2 Noise Subtraction for Object Detection

The initial basis for this work is a novel method to detect the presence of an object using the ratio of matched features to the total number of features in a target image. The target image is that of the object being searched for in a pair of scene images. By dividing the total matched features by the total features in the target image the correspondence ratio can be found. This normalises the number of features matched therefore different target images with varying numbers of features can be compared. For example, an image with five hundred features may have fewer matches to a scene than an image with two thousand features, but may have a higher correspondence ratio. The higher absolute number of

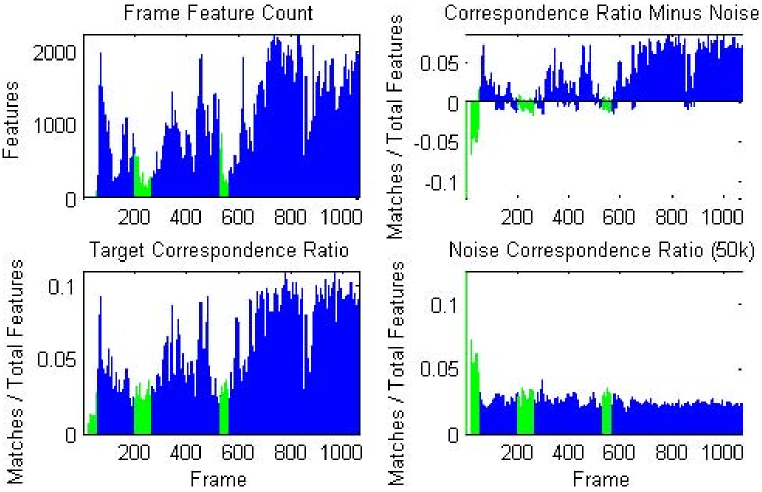


Fig. 1. The results of noise subtraction across 1062 frames of a video sequence. The top left graph shows the feature count for each frame. The top right hand graph shows the result of noise subtraction where the peaks indicate the presence of the object and the bottom left graph shows the target correspondence ratio for the object being identified (524 features) and the bottom right graph shows the false positive noise from a set of fifty thousand randomly collected features. Green (lighter) areas highlight those frames where the object is not present at all and these are shown to be negative on the top right, noise subtraction, graph.

matches in the second images may be noise (false positives) as the larger number of features available means more false positives will occur.

The technique uses the correspondence ratio for a large numbers of randomly collected features as a noise baseline for a particular scene. The features were collected automatically by randomly downloading large numbers of images from Flickr and applying SIFT to them. As these features are known to be random they are unlikely to match. This means that the ratio of matches indicates a level of matches that are statistically insignificant for an object that is being detected. As such, a ratio greater than this baseline of noise plus the average standard deviation can be deemed statistically significant (1σ) for detection. Tests have shown that using SIFT's default parameters has an average false positive rate of 0.024 and an average standard deviation of 0.007 for a random set of one million features. It has also been calculated that as few as ten thousand random features are enough to achieve these noise characteristics. This therefore means that on average a correspondence ratio greater than 0.031 is required for the number of matches to a scene to be deemed statistically significant.

By subtracting the noise correspondence ratio from the actual target correspondence ratio the data is automatically thresholded such that many false positives from the target to the scene will be ignored. Fig. 1 demonstrates this for a target image matched to 1062 frames of a video sequence where the object is present in most but not all of the frames.

3 FEWER: Feature Extraction and Weighting for Enhanced Recognition

Following this initial technique for subtraction of SIFT noise a second process has been developed which utilises the 3D stereoscopic image pairs of the target and scene to specify weighted feature matches to indicate confidence in their accuracy. This is called FEWER; Feature Extraction and Weighting for Enhanced Recognition. A pair of target images of the object that is being detected and a pair (or stream of pairs) of stereo images of a scene are used. Simply put, if a feature doesn't match well to its counterpart in a stereo pair the chances of it being stable are lower. The process has nine stages:

Extract SIFT Features. Extract the features from the target and scene stereo pairs as shown in Fig. 2.

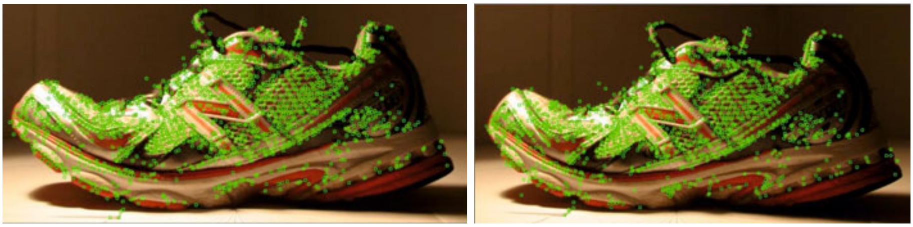


Fig. 2. A stereo pair of target images displaying the SIFT features extracted from them. There are 2176 in the left image and 2087 in the right image.

Calculate 3D Positions. For both the target and scene pairs a 3D point cloud is generated from the features as shown in Fig. 3.

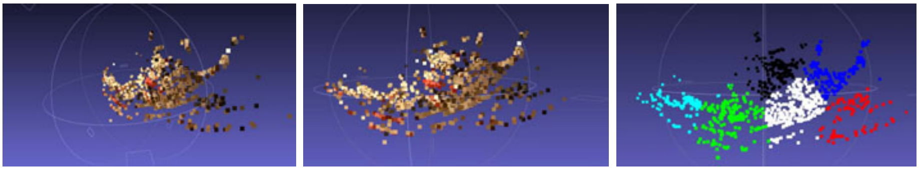


Fig. 3. The set of 3D feature positions generated from the stereo pair in Fig. 2 using the Bundler API [7]. The first two images show two different angles for the same data and the curvature of the shoe is clearly visible. This is a subset of the total features extracted from the original images and consists of 885 features. The right hand image shows the *type*3 features spatially clustered.

Cluster 3D Data. The 3D matched features are then spatially clustered in 3D space (using k-means [14]) to separate and label various 3D aspects of the scene. Clusters help differentiate between foreground and background objects.

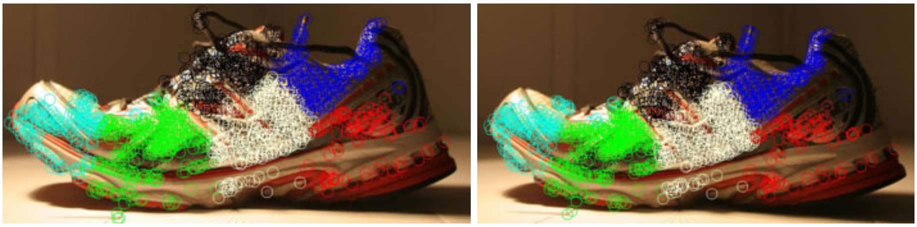


Fig. 4. The final set of clustered *type*2 and *type*3 features for the left and right images in a stereo pair. There are 1802 in the left image and 1782 in the right image.

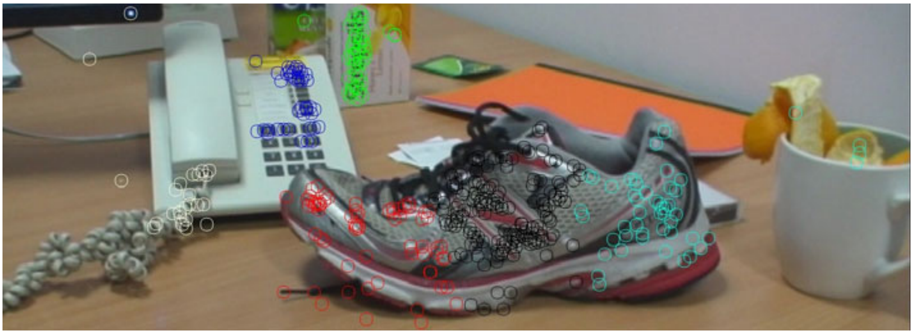


Fig. 5. The set of clustered features in a scene input image. The advantages of spatial clustering are clearer here as various objects have are roughly separated by the different clusters so as to provide more information when matching features.

Feature Labelling. Three different feature types are defined depending on their 3D and cluster properties. *Type*3 features are labelled by mapping the 3D features back to their 2D image locations for each image. *Type*3 features are those which have 3D information associated with them and therefore match to the other stereo image. To define *type*2 features a distance threshold is used to find other features near each of the *type*3 features and they are added to the clusters. These features are likely to be part of the same object as they are nearby but as they do not match to the other stereo image they can be considered less reliable. These are therefore labelled as *type*2 and a secondary cluster index is generated for them. The remaining features are then labelled as *type*1 and they do not have any cluster information relating to them.

Target to Scene Matching. Feature matching is performed for each target to scene combination; left target to left scene, left target to right scene, right target to left scene and right target to right scene. This is done using the nearest neighbour technique described by Lowe [13].

Initial Weighting. Each target image has its own set of weighting for matches to both of the scene images. Thus four sets of weightings are calculated. The initial weightings for each feature are given by which type they are and which type they match too. A *type*3 target to *type*3 scene match will have a larger initial weighting than a *type*3 target to *type*1 scene match. There are therefore nine possible combinations of matches each with their own weighting.

Type 3 Mismatches. The weightings are then adjusted by checking if matching pairs of *type*3 features from each target image match to similar positions in the scene images. Fig. 6 illustrates these cases. If the same *type*3 feature in both of the target images matches to different points in the scene the weighting is reduced as the likelihood of one or either being correct is reduced. The weighting is effected differently if the single scene feature is *type*3 or not *type*3.

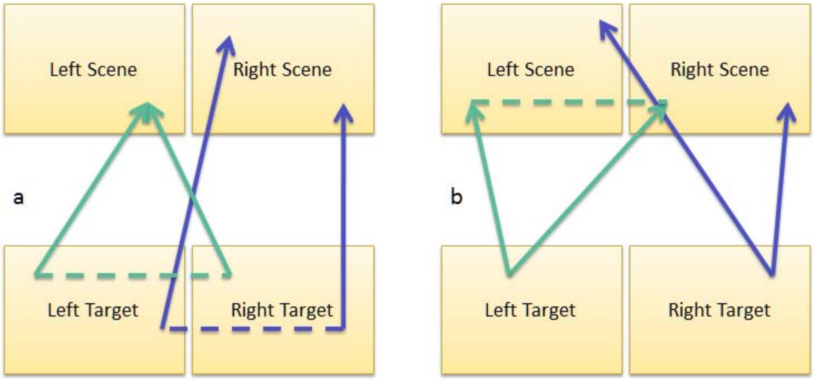


Fig. 6. This shows the two cases for *type*3 mismatches. Case *a* shows the correct (lighter) and incorrect (darker) matches from *type*3 features in the target images to any type of scene feature. Case *b* shows the correct and incorrect matches from the target image to the *type*3 scene features.

A secondary check is carried out for each target feature which matches to a *type*3 scene feature. If the feature matches to both corresponding *type*3 scene features then the weighting is increased. If a target feature matches a *type*3 scene feature and also matches a different feature in the other scene image then the weighting is reduced. There is no effect if the target feature matches one scene but not the other. Again the weighting is affected differently if the single target feature is *type*3 or not *type*3.

Cluster Weightings. The next stage is to adjust the weightings based on the 3D spatial cluster that a feature is in and how groups of features in the same cluster match. The basic hypothesis is that as more features in a target cluster match to a specific scene cluster the more likely it is that there is correspondence between these areas of the scenes. The confidence weighting is calculated as follows:

$$\text{confidence} = \frac{\text{signal}}{\text{noise}} \times \sqrt{\text{sample size}} \qquad (1)$$

where *signal* is the correspondence ratio from a target cluster to a scene cluster, *noise* is the correspondence ratio from the target cluster to every other scene cluster and *sample size* is the total correspondence ratio from the target cluster to all of the scene clusters. A confidence value is calculated for each target cluster to every scene cluster. This equation means that the sample size and the signal both have to be significantly large to generate a high confidence thus a low numbers of matches will not be statistically significant when calculating a feature's weighting. This confidence value is thresholded so that a high confidence cluster pair will result in a higher weighting for features which match between them. The boundaries and distribution of the clusters can affect the performance of this technique and as such there is no negative weighting for low confidence.

Threshold Matches. The weighting is normalised transforming its value into the range of 0 to 1. A threshold can now be applied to select a subset of the weighted feature matches.

Table 1. The stages used for extracting and weighting features with FEWER

Stage	**Output**
Extract SIFT Features	Sets of SIFT image features.
Calculate 3D Positions	Relative 3D positions of matched features.
Cluster 3D Data	Index of features indicating the cluster they are contained in.
Feature Labelling	Features labelled by type.
Target to Scene Matching	Indies indicating where features match to the scenes.
Initial Weighting	Weightings for each feature match.
Type 3 Mismatches	Updated weightings based on a disparity in matches.
Cluster Weightings	Updated weightings based on matches between clusters.
Threshold Matches	Set of matched features with weightings above a threshold.

4 Calculating Weightings from Noise

Values for the FEWER weighting adjustment stages described above have to be calculated to weight various characteristics of a matched feature. This is done by studying the noise properties for each stage using a set of stereo features know not to match correctly. By looking at the level of false positives for various feature match types ratios can be calculated which indicate how much more reliable one type of match is than another. The data describes how each type of match is affected by false positives. For the initial weighting stage the correspondence ratio for false positives for each match combination is calculated using large sets of random features. They are matched to videos which are known to contain no correspondence to the scene image. By obtaining the average correspondence ratio across the frames and adding the standard deviation it can be seen for the test data that *type*3 to *type*3 feature matches have a correspondence ratio 16 times less (0.64 / 0.04 from the full set of data listed in Tab. 2) than *type*1 to *type*1 thus the weighting reflects this directly. The weighting (w) is calculated as follows:

$$w = k\frac{1}{\bar{x}+\sigma} \times \frac{\text{relevant matched features}}{\text{total matched features}} \tag{2}$$

where $\bar{x}$ is the mean noise value across a sample, σ is the mean standard deviation of the noise and k is a scaling factor. The *relevant matched features* are the subset of the *total matched features* actually involved in the particular weighting process so that the weightings are scaled accordingly.

Table 2. Weighting values calculated from experimental data for different aspects of the weighting process. The left table shows the initial match weighting values and the right show the *type*3 mismatch weightings. T and S refer to Target and Scene.

	*type*1 S	*type*2 S	*type*3 S
*type*1 T	0.04	0.11	0.26
*type*2 T	0.06	0.41	0.39
*type*3 T	0.11	0.75	0.64

	*type*3 S	*type*3 T
correct *type*3	0.12	0.06
correct not *type*3	0.07	0.01
incorrect	-0.48	-9.3

The same process is used to calculate the weightings for the *type*3 mismatches where the number of false positives matches are used but as only the *type*3 features are involved the *relevant matched features* value reflects this. This incorporates a negative weighting for mismatches which have a relatively high cost as seen in Tab. 2.

For the cluster weightings, analysis has provided data on how well false positive matches cluster and what is the minimum level of cluster matching confidence required to occur beyond random chance. This allowed a cluster confidence threshold to be calculated using the same equation and a weighting for values greater than the threshold to be defined. This only relates to *type*2 and *type*3 features as *type*1 features are not clustered. The threshold was calculated to be 0.00015 and the weighting value added to matches greater than this threshold is 0.4 when using six clusters.

After these three stages the maximum possible weighting that can be achieved using the experimental data weightings is 1.36 and this value is used for normalisation.

5 Results

Following the weighting calculations based on over 2000 frames of video and over 20000 stereo target features, the system has been tested on different target and scene input data within a similar environment. The test involved a 2500 frame stereo video with a target object located within the sequence. Stereo images of the target objects are matched to each frame using the techniques described previously. The system outputs the four match images for each combination of target to scene matches with the matched features drawn using a heat map style colour coding. The colour changes linearly through RGB space from blue to green to red as the weighting increases.

Figs 7 and 8 shows examples of the coloured weightings as feature matches are deemed to be of higher or lower reliability. The images are consistent with the other frames in the sequence and show that incorrect matches are weighted lower.

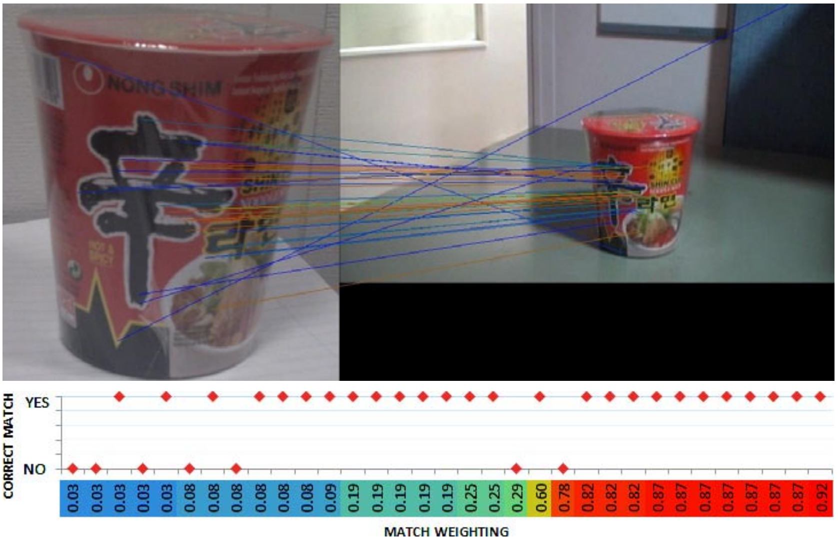


Fig. 7. A typical example of weighted feature matching displaying matches from the left hand target image to the left scene image. Some of the correct matches are green and red indicating higher weightings. The mismatched features in this scene have received low weightings and are coloured blue. The feature matches with low weightings can be removed by adjusting the weighting threshold which is set at 0 in these cases. The graph below shows the weightings for each of the 33 matched features and whether they match correctly.

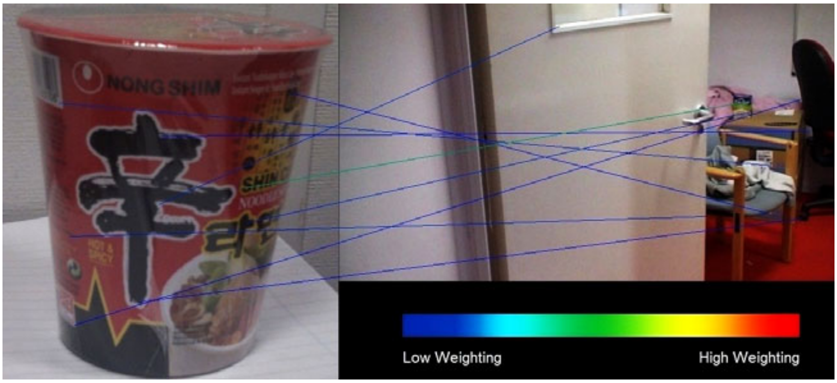


Fig. 8. A typical example of weighted feature matching displaying matches from the left hand target image to the left scene image. This shows false positives matches sucessfully being weighted with lower values.

Fig. 9 shows the correspondence ratio across the 2500 frames and the large peak indicates the location of the target. By adjusting the weighting threshold it is shown that the false positive count is reduced leaving many of the most reliable

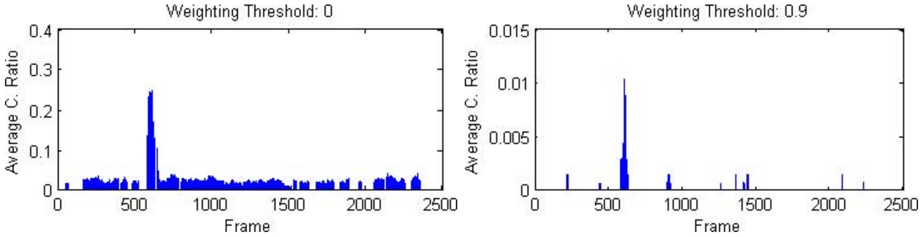


Fig. 9. This shows the correspondence ratio before and after applying a threshold on the feature weightings. The graphs are the mean of the four possible match scenarios (each target to each scene). The peak indicates the location of the object. The left graph shows the correspondence ratio when no threshold has been applied and the right graph shows what happens when a threshold of 0.9 is applied. This reduces the remaining correspondence ratio substantially but the features remaining are of a higher quality and fewer false positives are present across the video sequence.

features. The weighting threshold could be computed adaptively by analysing a set of known false positive feature matches in a similar manner to Section 2 and adjusting the weighting to minimise them.

6 Evaluation

FEWER has been shown to weight the features successfully. It relies on the probability of a feature type being a mismatch therefore, in some cases, incorrect matches can be weighted highly and vica-versa. Investigating how often this occurs will be future work. The weighting threshold provides a sliding scale between a small number of highly reliable matches and a large number of features including more unreliable matches.

The reason FEWER works is that $type3$ features are likely to be more stable than the other features as they correspond between the stereo images and are therefore known to match to a different view of the object. The SfM process [7] could be removed and normal SIFT matching used instead to generate $type3$ features. The SfM process has its advantages for clustering and background separation and is more discriminative when matching than just using SIFT as the matched features have to fit correctly to a 3D model not just match. The $type2$ features are more stable than $type1$ as the features are likely to exist on the objects that have been matched between the stereo objects due to their proximity to the $type3$ features and less likely to be background features. $Type1$ features are the least stable and have no extra properties associated with them. The difference between them is highlighted in Fig. 10.

FEWER allows the system to select a subset of features which are higher in confidence rather than just thresholding using the noise properties in Section 2 which has no indication of which features are likely to be correct. A combination

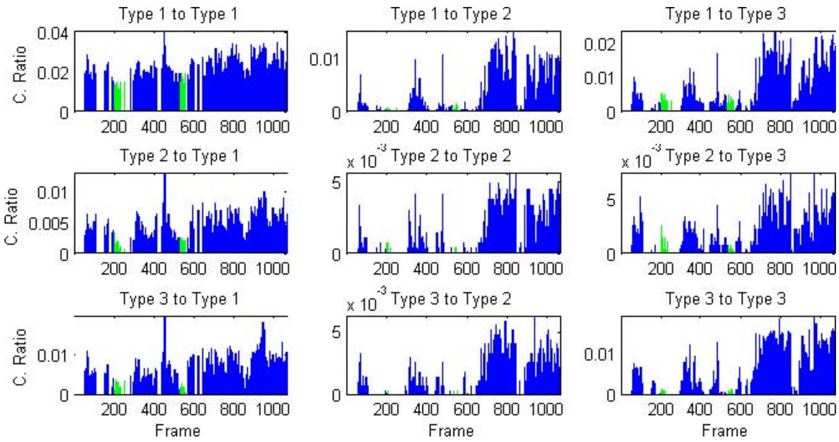


Fig. 10. These are the mean correspondence ratio graphs for the three feature types for matches from both target to both scene images. It can be seen that the *type*1 feature matches have fewer peaks and troughs and the green (lighter) areas, where the object is not present are harder to distinguish than for the *type*3 feature matches. They therefore resulted in lower weighting (see Tab. 2). For the random data used for calculating weightings in Section 4 these graphs are flatter with lower correspondence ratios. They display the random noisy correspondence ratio and give a minimum baseline for noise for each feature type.

of the noise thresholding for detection and FEWER could be used so that the computationally expensive weighting process is only applied to frames which are likely to contain the object to select the best matches.

7 Conclusion

The results of this work are promising and provide a technique for identifying and selecting the best feature matches. The results have shown examples of features being weighted to indicate which matches are correct and which are incorrect. The advantages of FEWER are clear as the detection process provides a higher confidence in the matches than standard SIFT matching alone. The system could result in lower data transmission rates as fewer matched features are selected.

Further development of the algorithm will involve data fusion to combine the four output images (left target to left scene etc.) into a single location mapped to a 3D model and superimposed on the 3D scene model. This will provide the user with a consolidated view of the output data to visualise where features match. Also, since the epipolar geometry is available, the weighting could possibly be improved at the matching stage by limiting the search region to a band around the epipolar lines. Comparison will be made to other methods for reducing the number of incorrect mathces using outlier detection methods such as RANSAC alone or the Hough binning used by Lowe [13].

References

1. Andriluka, M., Roth, S., Schiele, B.: Monocular 3D pose estimation and tracking by detection. In: IEEE CVPR, vol. (2), pp. 623–630 (2010)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
3. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and Recognition Using Structure From Motion Point Clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
4. Brown, M., Lowe, D.: Unsupervised 3D object recognition and reconstruction in unordered datasets. In: 3DIM, pp. 56–63. IEEE Computer Society (2005)
5. Coates, A., Ng, A.: Multi-camera object detection for robotics. In: ICRA, pp. 412–419 (2010)
6. Gould, S., Baumstarck, P., Quigley, M., Ng, A., Koller, D.: Integrating visual and range data for robotic object detection. In: ECCV M2SFA2 (2008)
7. Helmer, S., Meger, D., Muja, M., Little, J.J., Lowe, D.G.: Multiple Viewpoint Recognition and Localization. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 464–477. Springer, Heidelberg (2011)
8. Helmer, S.: Using stereo for object recognition. In: IEEE ICRA, pp. 3121–3127 (2010)
9. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. In: IEEE CVPR, vol. 2, pp. 2137–2144 (2006)
10. Laporte, C., Arbel, T.: Efficient discriminant viewpoint selection for active bayesian recognition. IJCV 68(3), 267–287 (2006)
11. Leibe, B., Cornelis, N., Cornelis, K., Van Gool, L.: Dynamic 3d scene analysis from a moving vehicle. In: IEEE CVPR, vol. 80, pp. 1–8 (2007)
12. Leibe, B., Cornelis, N., Cornelis, K., Van Gool, L.: Dynamic 3D Scene Analysis from a Moving Vehicle. In: IEEE CVPR 2007, pp. 1–8 (2007)
13. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
14. MacQueen, J.: Others: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, p. 14 (1967)
15. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(27), 1615–1630 (2005)
16. Quigley, M., Batra, S., Gould, S., Klingbeil, E., Le, Q., Wellman, A., Ng, A.: High-accuracy 3d sensing for mobile manipulation: Improving object detection and door opening. In: IEEE ICRA, pp. 2816–2822 (2008)
17. Sudderth, E., Torralba, A., Freeman, W., Willsky, A.: Depth from familiar objects: A hierarchical model for 3D scenes. In: IEEE CVPR, vol. 2, pp. 2410–2417 (2006)
18. Trajkovi, M., Hedley, M.: Fast corner detection. IVC 16(2), 75–87 (1998)
19. Whaite, P., Ferrie, F.: Autonomous exploration: Driven by uncertainty. IEEE TPAMI 19(3), 193–205 (1997)
20. Wojek, C., Roth, S., Schindler, K., Schiele, B.: Monocular 3D Scene Modeling and Inference: Understanding Multi-Object Traffic Scenes. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 467–481. Springer, Heidelberg (2010)

Verging Axis Stereophotogrammetry

Khurram Jawed and John Morris

Electrical and Computer Engineering, The University of Auckland, New Zealand
mjaw002@aucklanduni.ac.nz, j.morris@auckland.ac.nz

Abstract. Conventional stereophotogrammetry uses a canonical configuration in which the optical axes of both cameras are parallel. However, if we follow lessons from evolution and swivel the cameras so that their axes intersect in a fixation point, then we obtain considerably better depth resolution. We modified our real-time stereo hardware to handle verging axis configurations and show that the predicted depth resolution is practically obtainable. We compare two techniques for rectifying images for verging configurations. Bouguet's technique gives a simpler geometry - the iso-disparity lines are straight and the familiar reciprocal relationship between depth and disparity may still be used. However when the iso-disparity lines are the Veith-Muller circles, slightly better depth resolution may be obtained in the periphery of the field of view - at the expense of a more complex conversion from disparity to depth.

1 Preamble

Although the underlying geometry is well understood and mathematical models for verging axis stereophotogrammetry long published[1], the advantages of these configurations - discovered millions of years ago in the evolutionary process as animals learned to swivel their eyes in their sockets[2] - seem to have been substantially overlooked in favour of the trivially modeled canonical configurations in which the optical axes are parallel. Iso-disparity surfaces are also known as horopters[3]. The intersections between the horopters and the plane containing the optical centres and the fixation point are the Veith-Muller circles. Pollefeys *et al.*[4] analyzed these iso-disparity curves for different camera configurations. Olson *et al.*[5] studied the use of the horopter for active stereo heads. Here, we show that verging axis configurations lead to better depth resolution. Further, we implemented a real-time stereo system handling verging camera configurations in an FPGA. We report several experiments to validate the predicted positions and separations of the iso-disparity lines and demonstrate the enhanced depth resolution compared to a canonical stereo configuration.

Stereophotogrammetry systems usually capture 'raw' images from two cameras and then rectify them so that the images correspond to those taken by ideal pin-hole cameras - in a canonical configuration - with their axes parallel and perpendicular to the baseline joining the optical centres of both cameras[6]. This configuration has a significant advantage: scan lines of the rectified images are the epipolar lines so that the search for corresponding points in the two images may be constrained to the scan lines turning an $\mathcal{O}(n^2)$ search into a $\mathcal{O}(n)$

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 202–213, 2011.

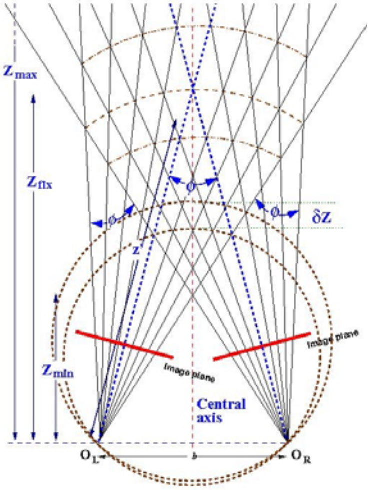


Fig. 1. Verging Axis Geometry showing two Veith-Muller circles. $O_{L|R}$ are the optical centres.

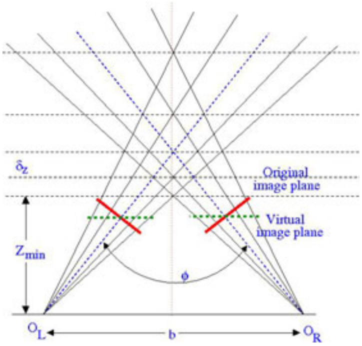


Fig. 2. Verging Axis Geometry using Bouguet's method[7] - the original image plane is transformed to the virtual one parallel to the base line and the principal points moved so that the optical axes intersect at the original fixation point

one. First, we show the theoretical benefits of verged axis systems (principally in enhanced depth resolution) and then show how our real-time stereo hardware was modified to gain these benefits.

If the cameras are deliberately verged, then we can use the same rectification procedures to convert the raw images to those taken by ideal cameras in the canonical configuration, but this loses the enhanced depth resolution of the verging configuration. We compare two techniques for rectifying verged camera images in ways that retain the improved depth resolution. We present some laboratory images of the same object taken with both configurations - empirically demonstrating the benefits and confirming the predicted benefit. Finally, the costs of both configurations were compared.

2 Stereo Geometry

A verging axis configuration is illustrated in Figure 1. For simplicity, we assume that two identical pin-hole cameras are rotated around an axis perpendicular to the baseline joining the optical centers of the two cameras so that the optical axes meet at a *fixation point* in the scene. We use capital letters, (X, Y, Z), for coordinates in a 'world' frame centred on the baseline midway between the optical centres of the two cameras, with its X-axis parallel to the baseline, its Z-axis lying in the same plane as the camera optical axes and its Y-axis perpendicular to the baseline. Lower case, (x, y, z) (with L or R subscripts as needed), is used for camera based coordinates and lengths. Both cameras are rotated about their

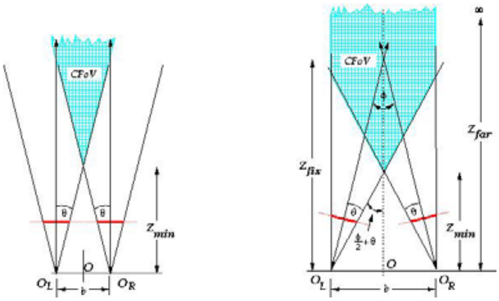


Fig. 3. Image plane use for a canonical configuration (left) *vs* a verging axis one (right): note the large areas outside the common field of view (CFoV) imaged in the canonical configuration

y-axes so that their optical axes intersect at an angle ϕ (the vergence angle) at the fixation point, $(0, 0, Z_{fix})$. Then

$$Z_{fix} = \frac{b}{2} \cot \frac{\phi}{2} \tag{1}$$

where b = baseline length. In the canonical configuration, the optical axes are parallel so $\phi = 0$ and $Z_{fix} = \infty$.

2.1 Depth Resolution

In stereo systems, depth is recovered from a pair of images by measuring the *disparity* or separation between pixels corresponding to the same scene point in the left and right images. In the verging axis configuration, the camera axes intersect at the fixation point in the scene. This point appears at the same position in both image planes and thus has disparity, $d = 0$. The loci of points with the same disparity are the Veith-Muller circles - see Figure 1. Considering only points along the central axis of the system, $X = 0, Y = 0$, the distance to points of disparity, d,

$$Z(d, \phi) = \frac{b}{2} \cot(\frac{\phi}{2} + \tan^{-1}(\frac{d}{2\lambda})) \tag{2}$$

where $\lambda = f/\tau$ (f = focal length and τ = pixel width) is the focal length in pixels.

Most stereo correspondence algorithms measure disparity in integral pixels only, so that the depth resolution at any point on the central axis is

$$\begin{aligned} \delta Z(d, \phi) &= Z(d, \phi) - Z(d-1, \phi) \\ &= \frac{b}{2}(\cot(\frac{\phi}{2} + \tan^{-1}(\frac{d}{2\lambda})) - \cot(\frac{\phi}{2} + \tan^{-1}(\frac{d-1}{2\lambda}))) \end{aligned} \tag{3}$$

Note that, in a verging axis system, unlike the canonical configuration, disparities may be negative: points with $Z > Z_{fix}$ will have $d < 0$.

Some practical constraints govern any stereo configuration. A practical correspondence algorithm will be able to handle disparities in some range, $d_{min} \leq d \leq d_{max}$, thus depth can only be measured in the area between $Z_{max} = Z(d_{max}, \phi)$ and $Z_{min} = Z(d_{min}, \phi)$ along the central axis ($X = 0$) and, in general, between the Veith-Muller circles for d_{max} and d_{min}.

To understand the increase in depth resolution, in Figure 1, observe the intersections of the rays projected through image plane pixels and the central axis. These rays intersect the line $X = 0$ at points of even disparity: thus the distance between any two intersections is roughly twice the depth resolution at that point. As the vergence angle increases, these gaps become smaller and depth resolution improves. We can also observe that the distance over which usable 3D data can be obtained, *i.e.* between Z_{min} and Z_{max}, shrinks as ϕ increases: this distance is divided into $d_{max} - d_{min} + 1$ measurable intervals, so depth resolution increases over the whole usable area. However, note that, in general, Z_{max} is no longer at infinity whereas $Z(d = 0, \phi = 0) = \infty$, so that the increased depth resolution is not without limitations. In practice this is rarely a problem, because the depth resolution, $\delta Z(d = 0, \phi = 0) = \infty$, is of little practical value.

Verging axis configurations also 'waste' less of the image planes of both cameras. Figure 3 shows wide regions of monocular points - for which no depth information can be derived. With a verging axis configuration, the full image planes of both cameras are used effectively.

In a canonical configuration, disparities are constant along straight lines parallel to the baseline, leading to the familiar relationship between depth and disparity:

$$Z = b\lambda/d \tag{4}$$

However in verging axis configurations, disparities are constant along the Veith-Muller circles (*cf.* Figure 1) leading to a more complex transformation, $d \rightarrow Z$[8], For corresponding pixels at $(u_{L|R}, v_{L|R})$ in the left and right images respectively:

$$\begin{aligned}
Z &= \frac{b}{\tan\left(\phi_L + \tan^{-1}\frac{u_L}{\lambda}\right) + \tan\left(\phi_R + \tan^{-1}\frac{u_R}{\lambda}\right)} \\
X &= \frac{b\tan\left(\phi_L + \tan^{-1}\frac{u_L}{\lambda}\right)}{\tan\left(\phi_L + \tan^{-1}\frac{u_L}{\lambda}\right) + \tan\left(\phi_R + \tan^{-1}\frac{u_R}{\lambda}\right)} - \frac{b}{2} \\
Y &= \frac{bv_L}{\lambda(\tan\left(\phi_L + \tan^{-1}\frac{u_L}{\lambda}\right) + \tan\left(\phi_R + \tan^{-1}\frac{u_R}{\lambda}\right))}
\end{aligned} \tag{5}$$

2.2 Rectification

Our first approach to rectification converts the original raw images to ones in which the 'scan' lines are these epipolar lines. Firstly, we remove distortion and align the images so that the optical axes intersect at $(0, 0, Z_{fix})$ (*i.e.* the cameras

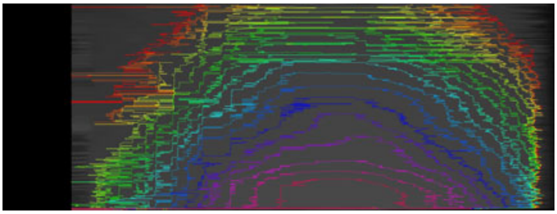


Fig. 4. Disparity map contours - flat panel roughly perpendicular to the system axis. Note that the regions of equal disparity are curved because the flat surface of the object intersects several Veith-Muller circles *cf.* Figure 1.

have been rotated around their y axes only[1]). The epipolar lines are now straight lines crossing the 'raw' images at an angle to the original scan lines (except for the scan line passing through the principal point).

Computing Epipolar Lines. The fundamental matrix, $\mathbf{F}$, was found using the eight point algorithm[9]. We identified corresponding pairs of epipolar lines for each image using $\mathbf{F}$[9]: for any point, p, in the left image, the corresponding epipolar line in the right image is $l' = \mathbf{F}p$. Similarly for a point p' in right image, the corresponding epipolar line in the left image is $l = \mathbf{F}^T p'$.

We now generate images in which the 'rows' are these epipolar lines: they can be fed directly to a correspondence algorithm used for a canonical configuration: it expects epipolar lines - in the canonical configuration, these are the same as scan lines. We simply changed the rectification lookup table so that it generated epipolar lines rather than scan lines. With this method, the depth resolution is the distance between adjacent Veith-Muller circles in Figure 1: it is given by Equation 3 along the central axis $(X = 0, Y = 0)$ of the system.

Figure 4 shows a disparity map obtained by this method. The viewed object is flat but equal disparity regions are curved as expected *cf.* Figure 1.

2.3 Bouguet's Method

An alternative method due to Bouguet[7] also preserves the enhanced depth resolution. It rectifies the two images into a canonical configuration and then re-projects them so that the optical axis meet at the fixation point. It computes a rectification matrix, $\mathbf{R}^{rect}$, that takes the epipole in the left camera to infinity. The rotation matrix, $\mathbf{R}$, computed from calibration, is split into two matrices, $\mathbf{R}_L$ and $\mathbf{R}_R$, rotating each camera by the same amount. From the original camera matrices, $\mathbf{M}_L$ and $\mathbf{M}_R$, rectified camera matrices are then computed $\mathbf{M}_L^{rect} = \mathbf{M}_L\mathbf{R}^{rect}\mathbf{R}_L$ and $\mathbf{M}_R^{rect} = \mathbf{M}_R\mathbf{R}^{rect}\mathbf{R}_R$. $\mathbf{M}_L^{rect}$ and $\mathbf{M}_R^{rect}$ are multiplied by

[1] This implies that small unintended rotations about camera x and z axes have been corrected.

projection matrices, with μ_{x_L} and μ_{y_L} set so that the two optical axis intersect at the fixation point - see Figure 2.

This method gives a slightly better depth resolution along the central axis $(X = 0, Y = 0)$ than the verging configuration in Figure 1, but slightly worse depth resolution for points on the periphery of the field of view, where the Veith-Muller circles get closer - see Figure 6. This changes the curved boundaries between regions of equal disparity to straight lines. Transforming disparity to depth uses the re-projection matrix:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & -\mu_{x_L} \\ 0 & 1 & 0 & -\mu_{y_L} \\ 0 & 0 & 0 & \lambda \\ 0 & 0 & -\frac{1}{b} & \frac{\mu_{x_L}-\mu_{x_R}}{b} \end{bmatrix} \tag{6}$$

where $(\mu_{x_{L|R}}, \mu_{y_{L|R}})$ is the optical center of the (left|right) camera.

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \frac{u}{2} \\ v \\ d \\ 1 \end{bmatrix} \tag{7}$$

The 3D coordinates are $(X/W, Y/W, Z/W)$. Depth resolution is now:

$$\begin{aligned} \delta Z(d)_{Bouguet} &= Z(d) - Z(d-1) \\ &= b\lambda(\frac{1}{d - (\mu_{x_L} - \mu_{x_R})} - \frac{1}{(d-1) - (\mu_{x_L} - \mu_{x_R})}) \end{aligned} \tag{8}$$

3 Implementation

3.1 Stereo Hardware

The real time stereo matching hardware uses Gimbelfarb's Symmetric Dynamic Programming Stereo algorithm[10]. It has a pair of Cameralink cameras attached directly to an Altera Stratix III FPGA connected to the host PC via an 8-lane PCIExpress bus. The FPGA removes lens distortion, rectifies the images and produces disparity and occlusion maps. It can compute dense disparity maps with 128 disparity levels at 30fps[11].

A checkerboard pattern was used for calibration[12].

For both rectification methods, all the corrections are combined into a single lookup table containing displacements for every pixel in the left and right rectified image. These lookup tables are reduced[11] and fed to the real time stereo matching hardware that produces the left and right rectified images and the disparity and occlusion maps.

4 Experiments

4.1 Experiment 1 - Stepped Target

For ground truth, we constructed a simple stepped target from Lego blocks, see Figure 5. Lego blocks are, of necessity, produced with the high dimensional accuracy needed to allow thousands of blocks to be used to build complex models. We measured a sample of blocks and confirmed that each block's dimensions were the same to within 0.1mm. Our test structure (Figure 5) has six steps each of two blocks and a height of 15.6 ± 0.1 mm.

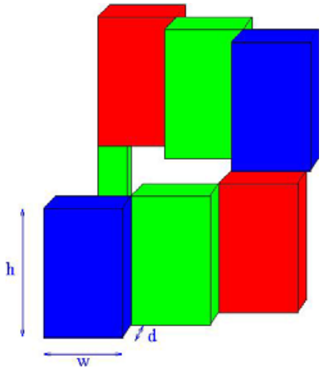


Fig. 5. Lego block structure h = 63mm, w = 40mm and d = 15.6mm

Disparity maps were acquired with both canonical and verging configurations with the target at various depths. The configurations were - canonical: b = 80mm, f = 9mm and $\phi = 0$ giving a predicted depth resolution from 9.7 mm to 993mm for disparity values from 126 to 12; verging: b = 427mm, f = 9mm and ϕ = 17.15^o, fixation point at 1400mm and predicted depth resolution from 1.6mm to 2.3mm for disparity values from 126 to 12. In the verging configuration, a longer baseline was used so that the target fills the field of view at approximately the same distance.

Disparity maps are shown in Figure 7. In every case, the expected disparity was observed and step depths were correct to within the predicted depth resolution for that disparity.

4.2 Experiment 2 - Sphere

In the second experiment, we used a ten-pin bowling ball: bowling balls must be precise spheres[2] so a ground truth can be derived from the geometry of a sphere. The ball was placed 600mm in front of the cameras and disparity maps were captured with both verging and canonical configurations. Configurations

[2] Round to within 0.010" (or 0.25mm in more widely accepted units)[13].

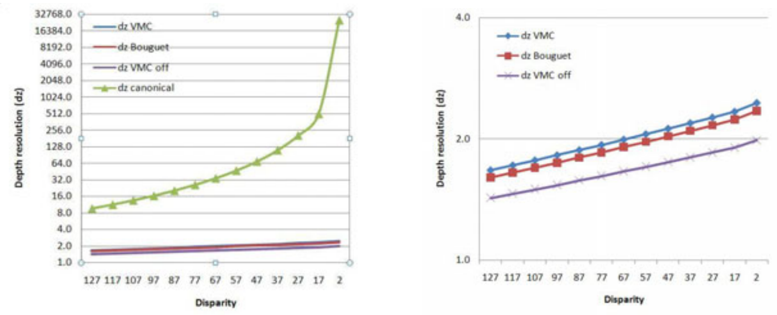
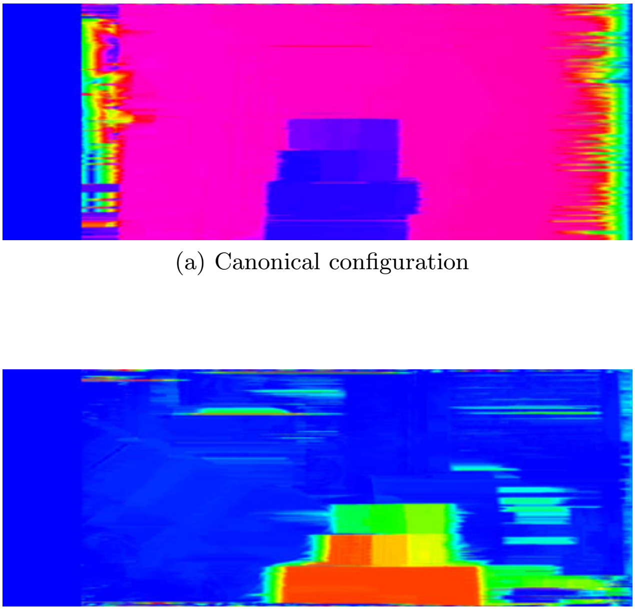


Fig. 6. Depth resolution for configurations of Experiment 1. dz VMC is depth resolution for a verging configuration (Figure 1), dz Bouguet is the depth resolution for Bouguet's method (Figure 2), dz canonical is the depth resolution for a canonical configuration and dz VMC off is the depth resolution for verging configuration but along the periphery of Figure 1. The left figure compares all configurations while the right figure compares verging configurations only at an expanded scale.

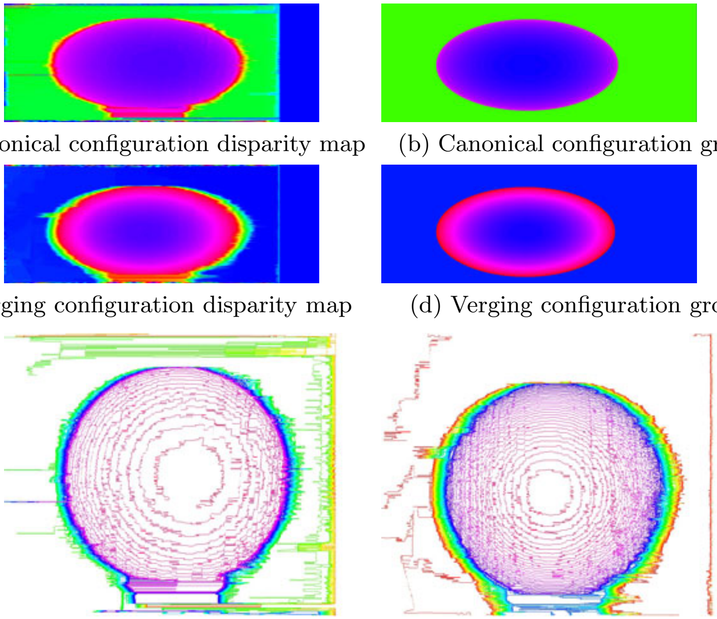


(a) Canonical configuration

(b) Verging axis configuration, $\phi = 17.15^o$

Fig. 7. Disparity maps for Lego block structure - note the increased number of disparity changes evident for the verging axis configuration

were - canonical $b = 38.7$mm, $f = 9$mm and $\phi = 0$, the depth resolution ranged from 4.9mm to 1.25m for a disparity range of 123 to 2; verging $b = 95.9$mm, $f = 9$mm and $\phi = 5.2^o$, depth resolution of 2.0mm to 5.5mm for a disparity



(a) Canonical configuration disparity map (b) Canonical configuration ground truth

(c) Verging configuration disparity map (d) Verging configuration ground truth

(e) Canonical configuration contours (f) Verging configuration contours

Fig. 8. Sphere experiment: contours on disparity maps. Note that SDPS produces a double-width disparity map[10], (a) through (d), leading to the apparently flattened images: when they are rescaled to the same width as the raw images, the contours are circles - see (e) and (f).

Table 1. Bowling ball matching performance

	canonical configuration	verging axis configuration
Threshold	% Bad pixels	% Bad pixels
0.5	70	68
1	44	37
1.5	22	15
2	10	7
RMS error (disparity units)	2.2	2.4

range of 123 to 2. and fixation point at 1055mm. The results of the experiments are summarized in Table 1 and the depth resolution plotted in Figure 9. The disparity maps and ground truth are in Figure 8.

4.3 Experiment 3 - Statue

The third experiment captured disparity maps of a complex object. This experiment demonstrates the increased depth resolution for a ‘real’ target. The configurations for this experiment were the same as those for experiment 2.

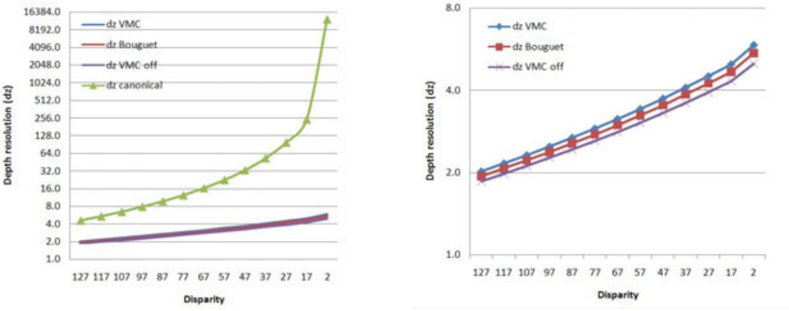


Fig. 9. Depth resolution for configurations used in Experiment 2. Labels are the same as for Figure 6.

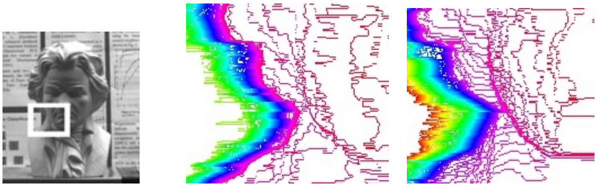


Fig. 10. Contours derived from disparity maps of the statue. From left to right: original raw image, canonical configuration contours and verging axis configuration contours. A small area of the disparity maps has been expanded to show the contour detail.

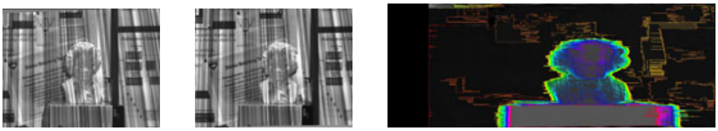


Fig. 11. System handling negative disparities: the fixation point is on the statue's nose, so that all points on the face and background appear to the left in the right image

Because there is no ground truth, Salmon[14]was used to find contours on disparity maps from both configurations. Contours on the verging axis disparity maps are more closely spaced due to the higher depth resolution. Note that, in the binocularly visible part of the face shown in Figure 10, there are 31 contours in the verging axis configuration disparity map compared to 13 - an increase in depth resolution of ~ 2.5.

4.4 Hardware Costs

In our FPGA hardware, rectification uses a lookup table which maps pixels in the desired configuration to actual image pixels. For either verging configuration, we simply compute a different lookup table and load it. Negative disparities are handled by trivial changes to the FPGA hardware - adjusting the length of the right pixel delay register - *decreasing* it for negative disparities. Figure 11 shows a pair of images with negative disparities and the contoured disparity map obtained. The total logic utilization was decreased by 4%.

4.5 Computation Costs

Rectification using the epipolar lines requires more complex computation to convert image coordinates to real-world coordinates. In software, conversion of a 1.5×10^6 entry disparity map to real world coordinates using Equation 7 takes 125ms, whereas using Equation 5 requires 325ms (2.0 GHz Pentium Dual Core). Either computation could be moved to the FPGA hardware leading to a negligible ($< 1ms$) increase in latency as pixels of the disparity map are streamed out of the correspondence circuit's back-track module[15].

5 Conclusion and Future Research

We have shown theoretically and verified experimentally that a verging axis configuration gives better depth accuracy. Verging axis configurations also generally produce a more useful common field of view. The depth resolution is essentially similar using either rectification approach. Some applications (*e.g.* collision avoidance, where we may only need a warning that a hazard has encroached an exclusion zone) can work with disparity data. Where conversion from disparities in pixels to world coordinates is required, Bouguet's rectification procedure is faster when the conversion must be performed in software on the host and adds slightly less latency if the hardware is used. The Veith-Muller circles give a slightly better depth resolution at the periphery of the common field of view but take longer to convert from disparity space to world space. In our FPGA system, rectification uses lookup tables, so there is no additional hardware cost or latency for a verging axis configuration: thus, the increased flexibility to design a more useful common field of view combined with superior depth resolution makes verging axis configurations preferable in practical configurations. Allowing the area behind the fixation point to be used (*i.e.* allowing negative disparities) produces a slightly smaller circuit by shortening the right pixel delay register and adds further flexibility in choosing the imaged region.

References

1. Maybank, S.: Theory of Reconstruction from Image Motion. Springer, Heidelberg (1993)
2. Meissner, G.: Beitrge zur Physiologie des Sehorgans. Leipzig, Engelmann (1854)
3. Aguilonii, F.: Opticorum Libri Sex philosophis juxta ac mathematicis utiles. Antwerp (1613)

4. Pollefeys, M., Sinha, S.: Iso-Disparity Surfaces for General Stereo Configurations. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004, Part III. LNCS, vol. 3023, pp. 509–520. Springer, Heidelberg (2004)
5. Olson, T.: Stereopsis for verging systems. In: CVPR 1993, pp. 55–60 (1993)
6. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision 47, 7–42 (2002)
7. Bouguet, J.Y.: Camera calibration toolbox for Matlab (1999)
8. Woods, A., Docherty, T., Koch, R.: Image distortions in stereoscopic video systems. In: Proceedings of the SPIE: Stereoscopic Displays and Applications IV, vol. 1915 (1993)
9. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press (2004)
10. Gimel'farb, G.L.: Probabilistic regularisation and symmetry in binocular dynamic programming stereo. Pattern Recognition Letters 23, 431–442 (2002)
11. Jawed, K., Morris, J., Khan, T., Gimel'farb, G.: Real time rectification for stereo correspondence. In: Xue, J., Ma, J. (eds.) 7th IEEE/IFIP Intl Conf on Embedded and Ubiquitous Computing (EUC 2009), pp. 277–284. IEEE CS Press (2009)
12. Bradski, G., Kaehler, A.: Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc. (2008)
13. United States Bowling Congress: Equipment Specifications and Certification Manual (2009)
14. Khan, T., Morris, J., Javed, K., Gimelfarb, G.: Salmon: Precise 3d contours in real time. In: Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2009, pp. 424–429. IEEE Computer Society, Washington, DC, USA (2009)
15. Morris, J., Jawed, K., Gimel'farb, G., Khan, T.: Breaking the 'ton': Achieving 1% depth accuracy from stereo in real time. In: Bailey, D. (ed.) Image and Vision Computing. IEEE CS Press, NZ (2009)

More on Weak Feature: Self-correlate Histogram Distances

Sheng Wang, Qiang Wu, Xiangjian He, and Wenjing Jia

Research Centre for Innovation in IT Services and Applications (iNEXT)
University of Technology, Sydney, Broadway 2007, Australia

Abstract. In object detection research, there is a discussion on weak feature and strong feature, feature descriptors, regardless of being considered as 'weak feature descriptors' or 'strong feature descriptors' does not necessarily imply detector performance unless combined with relevant classification algorithms. Since 2001, main stream object detection research projects have been following the Viola Jone's weak feature (Haar-like feature) and AdaBoost classifier approach. Until 2005, when Dalal and Triggs have created the approach of a strong feature (Histogram of Oriented Gradient) and Support Vector Machine (SVM) framework for human detection.

This paper proposes an approach to improve the salience of a weak feature descriptor by using intra-feature correlation. Although the intensity histogram distance feature known as Histogram Distance of Haar Regions (HDHR) itself is considered as a weak feature and can only be used to construct a weak learner to learn an AdaBoost classifier. In our paper, we explore the pairwise correlations between each and every histograms constructed and a strong feature can then be formulated. With the newly constructed strong feature based on histogram distances, a SVM classifier can be trained and later used for classification tasks. Promising experimental results have been obtained.

Keywords: Weak feature, Pairwise correlations, Histogram distances, SVM classifier.

1 Introduction

In computer vision research, it is widely recognized that good features are crucial for object detection tasks, there is abundant literature introducing state-of-the-art feature extraction algorithms [1][2][3]. Another research direction is the introduction of new object detection frameworks or improved feature extraction algorithm(s) [4][5]. In this paper, in addition to proposing a new feature based on correlate histograms, we are more interested in introducing a way to extract more information from an existing weak feature, we use the Histogram Distance of Haar Regions (HDHR) feature as an example.

In [4], the authors proposed the HDHR feature, the HDHR feature is defined as the intensity histogram distance between two adjacent Haar regions. Comparing with the simple Haar-like feature used by [6], the HDHR feature contains more

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 214–223, 2011.

information (hence should be able to better distinguish positive samples from negative samples) and can be calculated efficiently with the Integral Histogram framework proposed in [7][8]. An AdaBoost classifier is used in [4] to perform the object detection task of separating image regions that contains airplane from those regions which do not contain airplane.

In [9], the authors introduced the Shape Context feature descriptor, the shape context feature extraction algorithm is composed of three steps, the first step is to extract sample points from the edge map of the input image; the second step is to calculate the distance and orientation difference between the current sample point and every other sample point; the third step is to quantize those distances and orientation differences in to predefined number of bins. [9] is an early approach of feature extraction algorithms which are based on measuring object similarities with regard to certain distance metrics.

In [1], the authors introduced an approach to measure similarities between objects with a local descriptor, the descriptor is called Local Self-similarities (LSS). The LSS is based on matching internal self-similarities. That is, only the internal layout is correlated across images (or video sequences). Because the attributes for visual tasks (color, texture and illumination) within an image is relatively uniform compared to that of other images, exploring internal self-similarities can better capture the pattern of the visual entity. The LSS feature extraction process can be regarded as two steps, the first step is calculating correlation surface, this step is achieved by matching a smaller image patch from an image with a larger image region within the same image; the second step is translating the correlation surface into*a binned log-polar representation*, this step is similar to the final step of the Shape Context feature extraction. In [1], the CIE L*a*b space is used instead of the RGB color space to calculated the Sum of Squared Distances (SSDs) between patch colors. The LSS is a state-of-the-art feature descriptor based on self-similarity.

In [2], the authors introduced a new feature termed as Color Self-similarity (CSS), the CSS is based on the observation that objects such as a human do exhibit some structure in which colors are locally similar (*e.g.* the skin color of a specific person is similar on their two arms and face). In CSS, a positive sample (*i.e.* sample images which tightly bounds the object of interest) is first labeled with different semantic patches, such as arms, legs, upper body and background, then each semantic patch (of size 8 × 8 pixels) is used to measure the color similarity between the patch and the whole sample, the authors used HSV color space because it works best compared to RGB, HLS, CIE Luv, and etc. Each semantic patch will generate a similarity sample, in such similarity samples, the homogeneous region (for its corresponding similarity patch) will have a higher similarity score. Self-similarities between those similarity samples are then explored and utilized to construct a SVM classifier. In [2], the CSS is integrated with other features for object detection. It is one of the latest object detection approach using self-similarity measurement.

Motivated by the self-similarity feature being introduced in [1] and [2]. We propose a method that is capable of bring significant improvement over the saliency of the original weak feature such that a SVM classifier can be used to substitute the original AdaBoost classifier. Our feature extraction algorithm is composed of three steps, sub blocking, histogram binning, and correlating. Details will be given in Section 2.

Our contributions in this paper can be summarized as follows.

Firstly, by exploring its self-correlation, we transform a weak feature (HDHR) into a strong feature, we term it Correlation based Histogram Distance (COHD), this transformation is similar to the self-similarity features being proposed in [1]. As a strong feature, COHD enables the use of a SVM classifier for object detection, this saves a lot of time in comparison with having to train an AdaBoost classifier for the original weak HDHR feature.

Secondly, the newly proposed self-correlation feature based on histogram distances can be quickly calculated with the method proposed in [7], this is a precious computational advantage.

Thirdly, different from [1], which explores self-similarities from raw image level, we seek self-correlations from feature descriptor (*i.e.* Intensity Histogram) level, this can greatly reduce the computational cost and still well preserve the feature saliency.

The rest of this paper will be organized as follows, Section 2 introduces the formulation of a strong feature, we follow a typical object detection framework by replacing the original feature with the newly formed feature. Section 3 gives experimental results. Section 4 concludes this paper.

2 Weak Feature and Self-correlations

In this section, we will first introduce two types of weak feature, they are Density Variance feature and Histogram Distance of Haar Regions (HDHR) feature (neither of them can be directly combined with a SVM classifier for object detection task due to their weak saliency), then we introduce our proposed correlation feature derived from those two features mentioned above.

The Density Variance Feature was introduced in [5], such feature can be represented by

$$V_G = \frac{\sum_{i=1}^{n} |G_i - G|}{n \cdot G} \tag{1}$$

In (1), i is the index for the sub blocks as illustrated in Fig. 1, G is defined as the mean value of the gradient strength for the whole sample, and G_i is the mean value of the gradient strength for sub block i, n is the total number of sub blocks in a sample. In [5], the Density Variance feature was simply used as a global statistical filter to speed up the detection process for a license plate detector.

The Histogram of Haar Regions (HDHR) feature was first proposed in [4], the HDHR feature was introduced because of two reasons. Firstly, in order to differentiate two adjacent regions in a more suitable way, histograms provides more detailed information than classical Haar features. Secondly, Histograms can be computed linearly, which is a precious computational advantage. The HDHR feature descriptor is represented by

$$D(f,g) = \frac{\sum_{j=1}^{N} (f[j]-g[j])^2}{\sum_{j=1}^{N} (f^2[j]+g^2[j])} \tag{2}$$

In (2), D is defined as the Distance between the histogram $f[\cdot]$ and histogram $g[\cdot]$, as $f[\cdot]$ and $g[\cdot]$ each corresponding to a histogram constructed from image regions f and g, respectively. The number of bins in $f[\cdot]$ equals to the number of bins in $g[\cdot]$ and both equal to N, hence the distance calculation is a division of two summations over the bin index j. In [4], the HDHR feature was used together with AdaBoost supervised learning algorithm for airplane detection.

As mentioned in Section 1, our feature extraction method is composed of sub blocking, histogram binning, and correlating. Our sub blocking method was motivated by [5], our histogram binning method was motivated by [4], and motivated by [2], we use correlating to increase the feature salience.

In our approach, instead of considering the distance between two adjacent Haar-like Regions, we divide the sample image region into sub blocks of $p \times q$, in each sub block, a histogram can be constructed, hence the total number of histograms can be used to calculate D is $p \cdot q$. Given $p \cdot q$ histograms, we will consider the pairwise correlation between each pair of histograms, hence the total number of histogram distances can be measured is represented by

$$C^2_{p \cdot q} = \frac{(p \cdot q) \times (p \cdot q - 1)}{2} \tag{3}$$

Finally, the Correlation based Histogram Distance feature, we term it Correlation Histogram Distance (COHD) feature descriptor is represented by

$$\mathbf{S}_D = \{D(f,g)\} \tag{4}$$

which is a vector of length $C^2_{p \cdot q}$.

With COHD feature, an object detection framework can be easily constructed by train a Support Vector Machine (SVM) Classifier.

Moreover, we propose two variants based on different normalization schemes, the $L1-norm$ for COHD feature is represented by

$$E_1(f,g) = \sum_{j=1}^{N} |f[j]-g[j]| \tag{5}$$

The corresponding $L2-norm$ is represented by

$$E_2(f,g) = \sqrt{\sum_{j=1}^{N} (f[j]-g[j])^2} \tag{6}$$

In (5) and (6), the definitions for $f[\cdot]$, $g[\cdot]$, j and N are the same as those of (2).

By substitute D with E_1, the COHD $L1-norm$ feature descriptor is represented by

$$\mathbf{S}_{E_1} = \{E_1(f, g)\} \tag{7}$$

Similarly, the COHD $L2-norm$ feature descriptor is represented by

$$\mathbf{S}_{E_2} = \{E_2(f, g)\} \tag{8}$$

Details of the Correlation of Histogram Distance features ($i.e.$ S_D, S_{E_1}, and S_{E_2}) are illustrated in Fig. 1. In Fig. 1, f corresponding to the sub block from where histogram $f[\cdot]$ is constructed, and g corresponding to the sub block from where histogram $g[\cdot]$ is constructed.

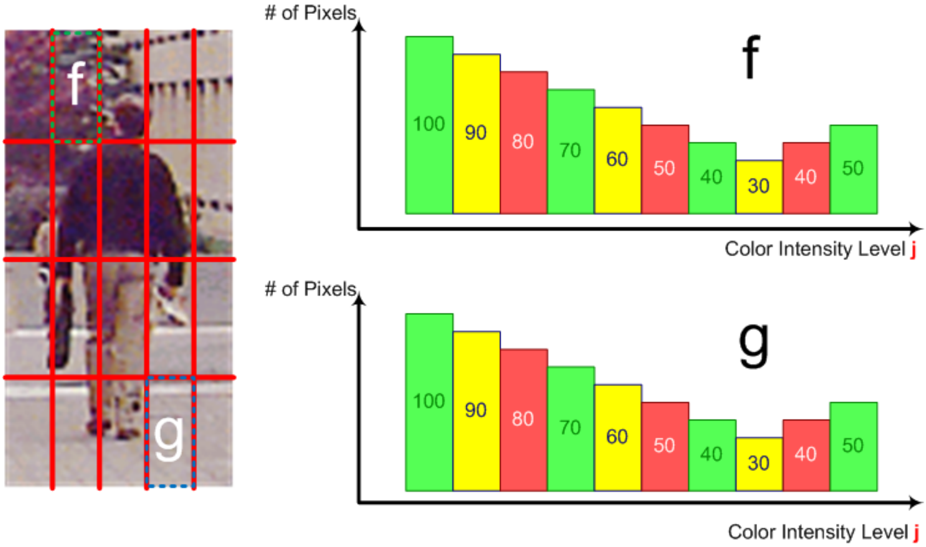


Fig. 1. Extracting Correlation of Histogram Distance features

The input image is first divided into $p \cdot q$ sub blocks, for each sub block f, a histogram $f[\cdot]$ can be obtained, $f[\cdot]$ is then compared with another histogram $g[\cdot]$ resulted from region g. The distance between $f[\cdot]$ and $g[\cdot]$ is one dimension of the $C^2_{p \cdot q}$-Dimensional feature vector.

3 Experimental Results

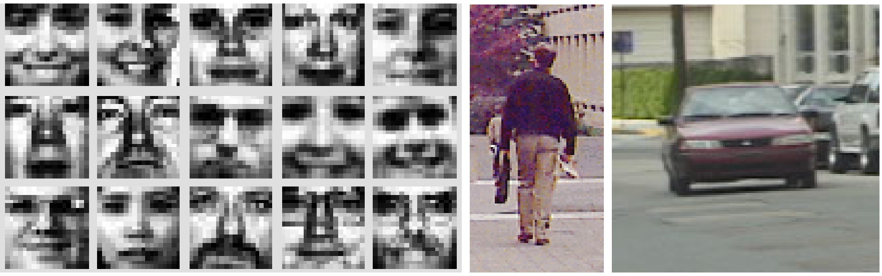
As one of the most representative strong feature, Histogram of Oriented Gradient (HOG) has attracted numerous attention of various researchers. As a result, we compare the descriptive power of HOG with our newly proposed correlation feature by replacing the HOG feature within the HOG and SVM framework with the correlation feature [10].

We use the MIT CBCL Dataset for our experiments, in particular, we evaluate the performance of the framework using Face, Human and Car [11][12][13]. The MIT CBCL Dataset is composed of four types of Data, they are, face, human, car, and scenario. More details of the Dataset can be found from Table 1.

Table 1. Details of the MIT CBCL Dataset

	Face	Human	Car
# of Positive Training Samples	2429	924	516
# of Negative Training Samples	4548	-	-
# of Positive Testing Samples	472	-	-
# of Negative Testing Samples	23573	-	-
Sample Size(Width×Height)	19 × 19	64 × 128	128 × 128

Some of the examples being used in our experiments can be found from Fig. 2.



(a) Face (b) Human (c) Car

Fig. 2. Some Examples from MIT CBCL Dataset

Detailed parameter settings can be found from Table 2.

As mentioned in [2], block normalization proven to be crucial, we use the same normalization scheme as provided in the MATLAB implementation of HOG and SVM framework by [10] to normalize the COHD feature descriptor.

A quantitative measure of the experimental results can be observed from Fig. 3. From Fig. 3, we can see that before sub block normalization, the newly proposed correlation feature based on HDHR can out perform HOG by approximately 4% on the MIT CBCL Face Dataset. However, the HOG feature remains extremely competitive on the MIT CBCL Human Dataset and MIT CBCL Car Dataset. Those results can be observed from Fig. 4 and Fig. 5, respectively. Yet our newly proposed feature (COHD with *L1-norm*) can achieve a detection rate of 97% at a false positive rate of approximately 2% on the Human dataset and 90% detection rate at 2% false positive rate on the Car dataset.

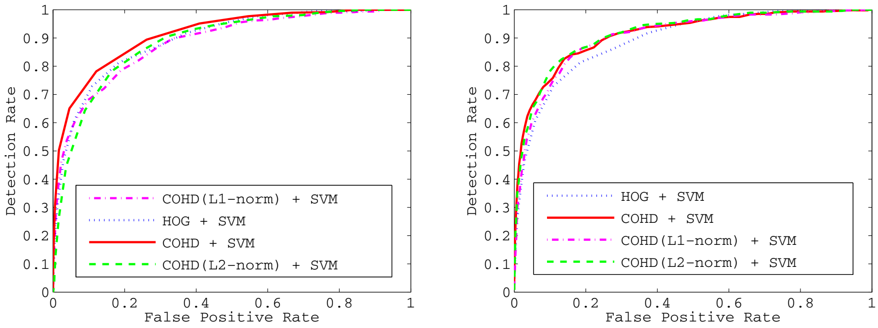
As we can see from Fig. 3b, Fig. 4b, and Fig. 5b, normalization can significantly improve the experimental results. The ROC curve for the human dataset

Table 2. Detailed parameter settings in our experiments[1]

	Face	Human	Car
# of Sub blocks (W×H)[2]	3 × 3	5 × 4	5 × 5
Scaled sample size (Width×Height)	19 × 19	32 × 64	32 × 32
# of Bins for COHD	32	32	32
# of Bins for COHD(*L1*)	32	32	32
# of Bins for COHD(*L2*)	32	32	32
# of Bins for HOG	9	9	9
# of Training Positive	2429	800	400
# of Training Negative	4548	1600	881
# of Testing Positive	472	124	116
# of Testing Negative	23573	195	160

[1] The negative samples for Human and Car Dataset was randomly cropped from background images which contains neither human nor car.

[2] W: the number of sub blocks in each row, H: the number of sub blocks in each column.



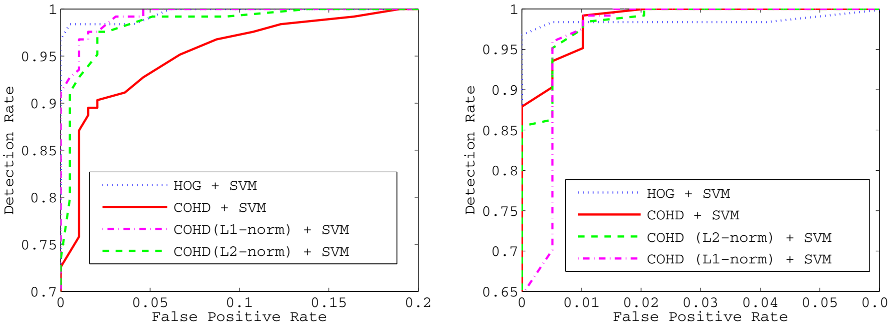
(a) Before Sub block Normalization

(b) After Sub block Normalization

Fig. 3. ROC Curves on MIT CBCL Face Dataset

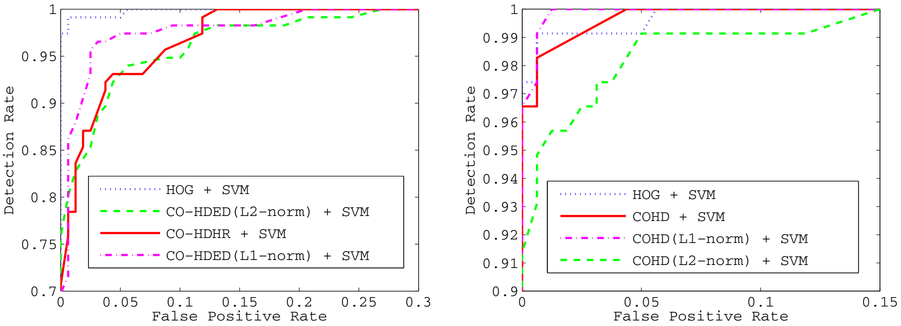
and car dataset is more rough than than that of the face dataset due to a smaller number of testing samples.

Those experimental results indicate that by exploring self-correlations, an original weak feature can be significantly improved to a strong feature, this approach of exploring intra-feature self-correlations is similar to the self-similarity features being proposed in [1][2][3], the difference is that the self-correlation is extracted from the feature descriptor level instead of the raw image data level, hence there will be some information loss to degrade the quality of the feature descriptor, but the computational complexity is also greatly reduced compared to that of self-similarity features and there is always a weight of balance between the computational cost and performance gain.



(a) Before Sub block Normalization (b) After Sub block Normalization

Fig. 4. ROC Curves on MIT CBCL Human Dataset



(a) Before Sub block Normalization (b) After Sub block Normalization

Fig. 5. ROC Curves on MIT CBCL Car Dataset

The proposed correlation method does not require matrix convolution during the feature extraction process, comparing with HOG, which needs gradient magnitude computation and arc tangent computation, the feature extraction process is much simpler. Although to extract Haar-like feature is also very simple, the computational cost (especially time complexity) for AdaBoost training is very high, this computational advantage is especially important for devices with limited computational power, such as wireless sensors.

The computational cost (in terms of time complexity) to measure a pairwise histogram distances for a detection window that is partitioned into k sub windows is $\frac{k\times(k-1)}{2}$. Without using Integral Histogram, the computational cost needed to calculate the histogram feature of a detection window of size $n \times n$ is $O(n^2)$, the Integral Histogram can reduce this cost to $O(1)$. As reported by [3], to calculate the LSS descriptor for one pixel with patch size $\omega \times \omega$ and block size $N \times N$ requires $N^2\omega^2$ operations, the authors for [3] also mentioned that although Fast Fourier Transform(FFT) can speed up the process with $3N^2logN^2 + N^2$ operations, the speed up is marginal as $N > \omega$.

In our experiments, we also compared the execution speed of the COHD feature extraction algorithms with that of the HOG feature extraction algorithm on the same platform, details are listed in Table 3. For the implementation of HOG feature extraction, we use the code provided by [10]. Depending on each particular sample, the speed of feature extraction varies, hence we compare the total time needed to convert the entire training dataset to corresponding feature descriptors. Details about each dataset is given in Table 2. Using Matlab 2009b with a Windows XP(32bit) environment, on a computer with 3.16GHz CPU and 3.25GB of RAM, we obtained the results in Table 3.

Table 3. Speed Comparison for Feature Extraction

	Face	Human	Car
HOG [10]	16.42 seconds	10.88 seconds	11.70 seconds
COHD	11.98 seconds	7.85 seconds	6.28 seconds
COHD(*L1*)	11.99 seconds	7.84 seconds	6.28 seconds
COHD(*L2*)	12.04 seconds	7.98 seconds	6.38 seconds

4 Conclusion

In this paper, we have proposed a self-correlation method to improve the saliency of a weak feature, by dividing the detection window into sub blocks, we have proposed three different normalization schemes for self-correlated features derived from intensity histograms. The experimental results on MIT CBCL Dataset proved that those self-correlated features can dramatically increase the feature saliency. In particular, for MIT CBCL Face Dataset, the self-correlated feature outperform one classical strong feature object detection framework. However, this method is not limited to one particular type of feature, other weak features can be enhanced by this self-correlation method as well.

References

1. Shechtman, E., Irani, M.: Matching Local Self-similarities across Images and Videos. In: Proc. CVPR, Minneapolis, pp. 1–8 (2007)
2. Walk, S., Majer, N., Schindler, K., Schiele, B.: New Features and Insights for Pedestrian Detection. In: Proc. CVPR, San Francisco, pp. 1030–1037 (2010)
3. Deselaers, T., Ferrari, V.: Global and Efficient Self-similarity for Object Classification and Detection. In: Proc. CVPR, San Francisco, pp. 1633–1640 (2010)
4. Perrotton, X., Sturzel, M., Roux, M.: Automatic Object Detection on Aerial Images Using Local Descriptors and Image Synthesis. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 302–311. Springer, Heidelberg (2008)
5. Zhang, H., Jia, W., He, X., Wu, Q.: Learning-Based License Plate Detection Using Global and Local Features. In: Proc. ICPR, Hong Kong, pp. 1102–1105 (2006)

6. Viola, P., Jones, M.: Rapid Object Detection Using A Boosted Cascade of Simple Features. In: Proc. CVPR, Kauai, pp. 511–518 (2001)
7. Porikli, F.: Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces. In: Proc. CVPR, San Diego, pp. 829–836 (2005)
8. Kovesi, P.: University of Western, Australia, http://www.csse.uwa.edu.au/
9. Belongie, S., Malik, J.: Matching with Shape Contexts. In: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries, Hilton Head, pp. 20–26 (2000)
10. Ludwig, O., Delgado, D., Goncalves, V., Nunes, U.: Trainable Classifier-Fusion Schemes: An Application to Pedestrian Detection. In: Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, pp. 432–437 (2009)
11. Weyrauch, B., Huang, J., Heisele, B., Blanz, V.: Component-based Face Recognition with 3D Morphable Models. In: Proceedings of the First IEEE Workshop on Face Processing in Video, Washington, D.C, pp. 85–89 (2004)
12. Papageorgiou, C., Evgeniou, T., Poggio, T.: A Trainable Pedestrian Detection System. In: Proceedings of the IEEE International Conference on Intelligent Vehicles, Stuttgart, pp. 241–246 (1998)
13. Oren, M., Papageorgiou, C.P., Sinha, P., Osuna, E., Poggio, T.: Pedestrian Detection Using Wavelet Templates. In: Proc. CVPR, San Juan, pp. 193–199 (1997)
14. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Proc. CVPR, San Diego, pp. 886–893 (2005)

Mid-level Segmentation and Segment Tracking for Long-Range Stereo Analysis

Simon Hermann[1,⋆], Anko Börner[2], and Reinhard Klette[1]

[1] The *.enpeda..* Project, Department of Computer Science
The University of Auckland, New Zealand
[2] DLR (German Aerospace Center), Berlin-Adlershof, Germany

Abstract. This paper presents a novel way of combining dense stereo and motion analysis for the purpose of mid-level scene segmentation and object tracking. The input is video data that addresses long-range stereo analysis, as typical when recording traffic scenes from a mobile platform. The task is to identify shapes of traffic-relevant objects without aiming at object classification at the considered stage. We analyse disparity dynamics in recorded scenes for solving this task. Statistical shape models are generated over subsequent frames. Shape correspondences are established by using a similarity measure based on set theory. The motion of detected shapes (frame to frame) is compensated by using a dense motion field as produced by a real-time optical flow algorithm. Experimental results show the quality of the proposed method which is fairly simple to implement.

1 Introduction

The classification of traffic-relevant objects (e.g. vehicles, bicyclists, pedestrians, or traffic signs) is a common goal in vision-based *driver assistance systems* (DAS). If an object is potentially dynamic, then it is important to understand its current state (e.g., currently static, or moving with a particular trajectory). This task requires solutions for scene segmentation, object detection and tracking, and eventually also for object classification.

Monocular or stereo vision, LIDAR, or infrared cameras are sensors considered for solving such tasks. We apply one pair of grey-level cameras for stereo vision that records the space in front of the *ego-vehicle* (i.e. the vehicle the cameras operate in). Given a 3D world coordinate system, stereo analysis provides depth information about the traffic scene in form of a cloud of 3D points. After ground manifold estimation and possibly its removal [14,22], subsequent tasks are free-space estimation [2], obstacle detection using occupancy grids [16], or object segmentation by point clustering [17]. A tracking process involves motion analysis [4,15], e.g. using particle filters [6,17].

A recent approach [20] uses disparity information for the purpose of real-time 3D scene flow computation. Object motion in the 3D world is calculated using ego-motion estimation [1]. A motion likelihood is assigned to each pixel and a binary graph-cut segmentation algorithm identifies independently moving objects [21].

[⋆] The first author thanks the German Academic Exchange Service (DAAD) for financial support.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 224–235, 2011.

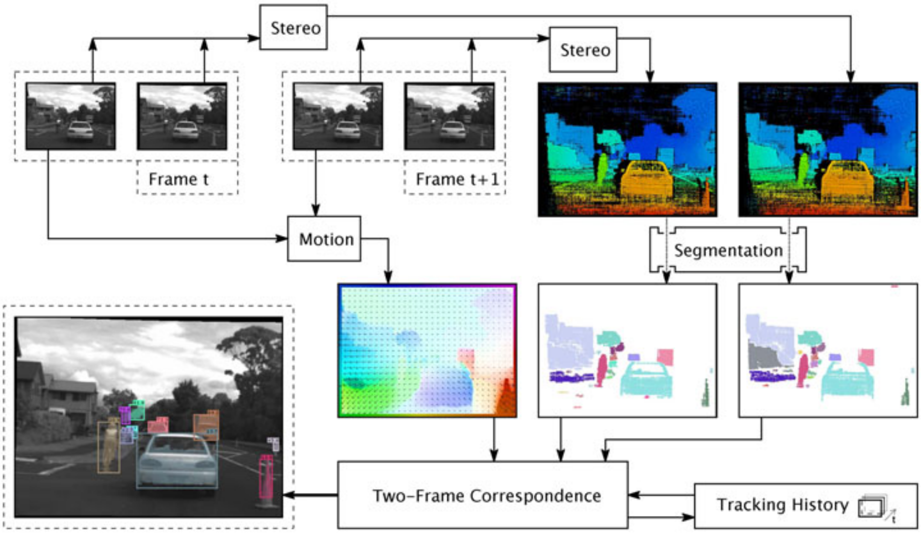


Fig. 1. Upper left: input data at t and $t + 1$. Upper right: disparity maps. Middle and lower right: optical flow map and segmentation results. Lower left: a final result showing enclosing rectangles (the *bounding boxes*) for tracked segments.

This paper proposes a new technique of mid-level scene segmentation and segment tracking. The key feature of our approach is that both, segmentation and tracking are entirely performed within the disparity data generated by dense stereo algorithms. Our approach has the advantage that DAS tasks are solved before reprojecting disparity values into the 3D world. This eliminates one source of errors, and supports the design of 2D algorithms for segmentation and tracking on disparity maps. The resulting implementation, as outlined below, is of great simplicity.

Disparity-based segmentation is already extensively used for pedestrian detection [7,27] by identifying back- and foreground areas in an image [27], before applying intensity-based segmentation.

Ground-manifold approximation [14] and scene-flow computation [20] can be solved in disparity space.

In [11] the problems that occur by tracking 3D data is highlighted and the authors argue on the benefits of performing these task in disparity space. However, they evaluate their proposed method only on synthetic data and do not provide any result image. In [24] segmentation and and tracking is performed in disparity space, but with a different approach.

To the best knowledge of the authors, the proposed method constitutes a novel approach for scene segmentation and tracking in disparity space. Our implementation was evaluated on three real-world traffic sequences of 400 or 250 stereo frames. Segmentation results are also compared with available ground truth.

At this stage we purposely do not consider intensity information from the input images for segmentation. The reason is that we want to highlight the segmentation and tracking quality that is possible by exploiting disparity information only. Of course, future work should incorporate intensity information into the proposed segmentation and tracking process.

The Proposed Method. Figure 1 sketches the workflow of our approach. The input are two stereo pairs at times t and $t+1$. A stereo matcher computes for each stereo pair a dense disparity map. (The left image is our reference frame). Each disparity map is segmented according to a three-step segmentation process:

First, we post-process the disparity map with a mode-based filter that removes 'noisy disparities' (e.g. in occluded areas, or in irrelevant areas such as the sky). *Second*, a road or ground manifold is estimated and subtracted from the stereo map. This prevents objects from being connected by similar disparities at road level. *Third*, the resulting disparity map is segmented by employing a simple region-growing algorithm.

The segmentation process is described in detail in Section 3. Results during the segmentation process are shown in Fig. 1. Final patches are very similar in shape and location at t and $t+1$. A human would easily identify corresponding patches in segmentation maps f_t and f_{t+1}. A set-theoretical metric is applied for quantifying correspondences between 2D patches.

The metric encodes the ratio between overlap and total area of both patches, and thus also the similarity in 2D shape. The latter can be assumed because it is reasonable to neglect roll or tilt of objects in vision-based DAS.

We expect invariance of projected 2D object shape between subsequent frames (at least for rigid objects) recorded at 25 Hz or more, but aim at handling (minor) changes in object size due to varying distances to the ego-vehicle, and translational changes in positions due to recording highly dynamic scenes. To compensate for translational changes we calculate a dense motion field using a real-time state-of-the-art optical flow algorithm. Calculated motion vectors are used to shift pixels (of an object) from segmentation map f_t into new positions in f_{t+1}. So far we do not rescale a patch before we apply the metric because different objects cannot occupy the same image region, and a change in size did not appear to be very crucial for identifying corresponding patches.

After correspondence analysis, a temporal filter calculates the size of the current patch. This size is used for rescaling the bounding box. Correspondences between patches define the tracking history.

Outline of the Paper. Section 2 presents the used stereo and motion analysis algorithms with comments about their parametrization. Section 3 explains ground manifold estimation, the mode filter used for stereo post-processing, and the proposed segmentation for one stereo pair into objects (or patches). Section 4 establishes correspondences between patches in subsequent frames using a known shape metric, and describes the proposed tracking mechanism that works on image sequences and uses shape priors. Section 5 summarizes experiments about segmentation and tracking. Section 6 concludes.

2 Stereo and Motion Analysis

Semi-global Stereo Matching. For generating a dense disparity map D, we follow the original semi-global stereo matching algorithm [9] that minimizes the energy

$$E(D) = C(D) + S(D) \tag{1}$$

where $C(D)$ refers to the dissimilarity or data cost and $S(D)$ to the smoothness cost which incorporates a first-order data prior.

For the data term we apply the census cost function which calculates the Hamming distance of two binary signature vectors which are assigned to corresponding pixels; see the census transform in [25]. It has been shown [10] that this function is very descriptive and robust, even under strong illumination variations, which is crucial for real-world applications. In our implementation we use a 9×3 window as we work on a 32-bit machine and favour a stronger data contribution along the epipolar line.

The semi-global smoothness constraint integrates multiple optimal 1D energies along different accumulation paths using a dynamic programming approach. Since the number of paths is very limited (usually not more than eight paths) it is referred to as semi-global matching (SGM).

Our implementation uses four accumulation paths (up, down, left, right). Subpixel accuracy is obtained using equiangular interpolation [18]. To enforce uniqueness, two disparity maps are calculated and a left-right consistency check is performed. A disparity passes this test if corresponding disparities do not deviate by more than 0.7 disparity levels. The consistency check is performed to invalidate occluded areas and to remove ill-defined disparity values. Figure 2 shows a calculated disparity map. The colour code runs from "hot" (large disparities) to "cold" (small disparities).

Dense Motion Estimation. We considered methods published in [4,26] for dense motion field calculation. Both methods are based on the total variation approach by Horn-Schunck, but instead of minimizing a global energy based on the L_2 norm (as in the original work), they minimize an energy that uses the L_1 norm as data term. Although performance between both methods is rather similar, the numerical schemes of both methods are quite different. While [4] uses a fixed point iteration procedure to solve Lagrange equations, [26] employs a duality-based approach. Both algorithms are suitable for parallel implementation and can achieve real-time performance. The implementation reported in this paper follows [4]. Figure 3 shows a result for this algorithm. The rectangular frame also shows the used colour key.

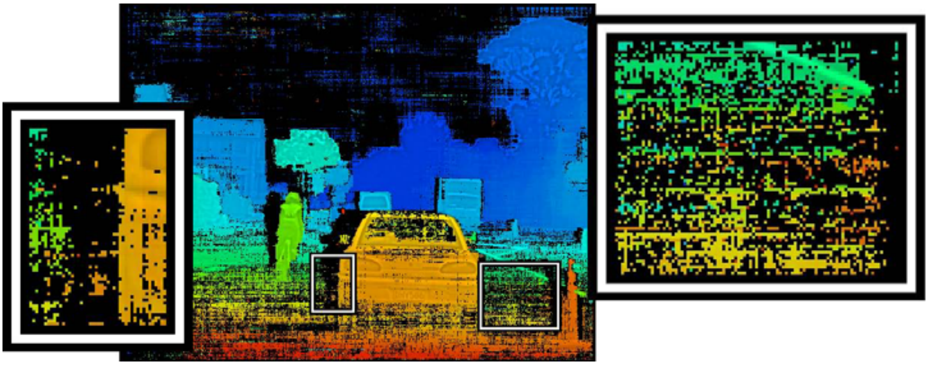


Fig. 2. Result of the SGM algorithm with the proposed settings. Enlarged windows show cases to be processed by our mode filter.

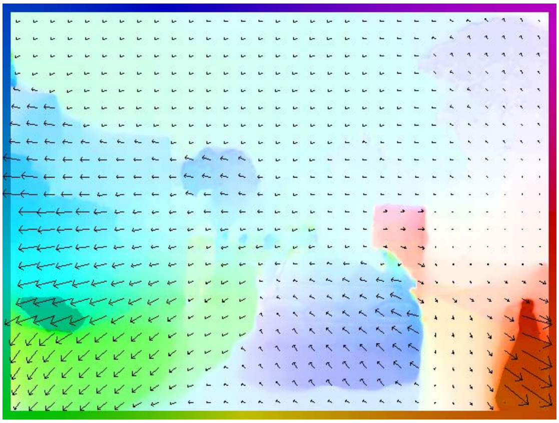


Fig. 3. Result of the used pyramidal optical flow implementation following [4]

3 Segmentation in Disparity Space

Mode Filtering. [9] proposed to apply a median filter to disparity maps before performing the consistency check. The filter removes disparity outliers and performs edge-preserving local smoothing of disparity values. As a result, more pixels pass the consistency check, thus increasing the denseness of the final disparity map.

However, for the purpose of segmentation, denseness is not the primary goal. On the contrary, enhancing occluded areas or invalidating disparities close to object boundaries helps in the stereo segmentation process. Since occluded areas have a low disparity denseness, a filter that supports the segmentation process should identify this characteristic and invalidate remaining pixels in that area. Additionally, the filter should invalidate image regions where disparity information is rather undefined. Such areas are quite often affected by noise. The enlarged window in Fig. 2, left, shows disparity values which correspond to the disparity level of the car; they are propagated into the occluded area. The enlarged window on the right shows a noisy disparity region on the road.

Intuitively, we are looking for a filter which generates both, a disparity value that is statistically dominant within a local neighbourhood, and an index that indicates the support or likelihood for this dominant value. If the corresponding disparity value of a neighbourhood is close enough to the identified dominant disparity, and the support is sufficiently high, then the disparity value should remain unchanged. Otherwise it should be discarded.

Computing the mode of a neighbourhood fulfils both needs. The mode identifies one dominant value that corresponds to the highest occurrence of a value in a domain with a fixed number of elements. The ratio of occurrences to the number of elements in the domain serves as index for the support. Since we work on sub-pixel disparity levels we need to count occurrences of disparities within intervals. The centre of the interval that contains the most disparities represents the dominant disparity value η.

Let $\mathcal{N}_p$ be the pixel neighbourhood of a pixel p. The ordered sequence $\mathcal{S}$ of disparity values within this neighbourhood is $\mathcal{S} = \langle s_0, s_1, ..., s_n \rangle$. We define for $\xi \geq 0$ and each disparity s_i in $\mathcal{S}$ a subset

$$\mathcal{I}_i = \{s : \ s \text{ in } \mathcal{S} \ \wedge \ s_i \leq s < s_i + \xi\} \tag{2}$$

of all disparity values in $\mathcal{N}_p$. Assume that the maximum

$$c_m = \max_{i=0,1,\ldots,n} \operatorname{card}(\mathcal{I}_i) \tag{3}$$

is for set $\mathcal{I}_m$. The centre

$$\eta = s_m + \frac{\xi}{2} \tag{4}$$

of the corresponding interval defines the mode of the neighbourhood $\mathcal{N}_p$. The support ν is the ratio of the number of elements in $\mathcal{I}_m$ to the number of elements within the neighbourhood $\mathcal{N}_p$:

$$\nu = \frac{c_m}{\operatorname{card}(N_p)} \tag{5}$$

Let D_p be the disparity value at pixel p in disparity map D. We define our *mode filter* by the following rule:

$$D_p = \begin{cases} D_p & \text{if } |D_p - \eta| < \xi \wedge \nu > \psi \\ invalid & \text{otherwise} \end{cases} \tag{6}$$

This filter has two input parameters. ξ defines the disparity range of considered intervals and ψ defines the percentage how many disparities need to be in the mode interval. Choosing $\psi > 0.5$ avoids ambiguous cases when a maximum cardinality is taken by more than just one set $\mathcal{I}_m$. Figure 4 shows the result of this filter.

Ground Manifold Removal. It is common practice to approximate the ground manifold by a planar surface. A simple and fast, yet very robust method [14] uses a *v-disparity map* defined as follows:

Let $\{0, 1, \ldots, d_{max}\}$ be the disparity range of an $M \times N$ disparity map D in the *ij*-plane. This 2D array is projected into a 2D v-disparity array V in the *dj*-plane:

$$V(d, j) = \sum_{i=1}^{M} \Psi(D_{ij} = d) \tag{7}$$

for $0 \leq d \leq d_{max}$ and $1 \leq j \leq N$, where $\Psi(true) = 1$ and equals zero otherwise. In [14], a dominant straight line is detected by Hough transform after binarization of V.

We approximate a dominant straight line by linear regression, where those points are iteratively discarded whose residues lie above the convergence threshold. For the initial point cloud, one representative is chosen for each disparity d, namely the second largest[1] j-value to where a value is projected. This line has the property of ‘rising from below’ towards the scattered points in the v-disparity map and it is ‘resting’ at a stable position. Disparities close to this ground manifold are removed prior segmentation.

[1] The origin of map V is in the top-left corner.

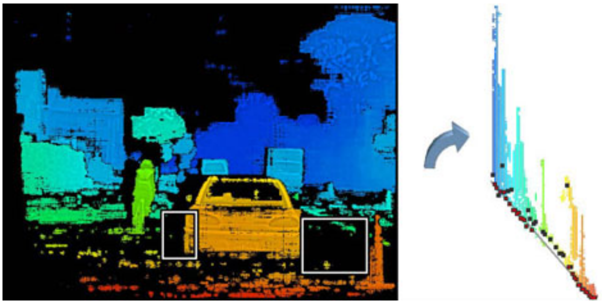


Fig. 4. Left: Disparity map after applying the mode filter. Right: The corresponding v-disparity map with calculated 'lower straight envelope'.

Figure 4 shows a result for our regression method. For better visibility, we just project the disparity values and not their frequencies. The initial point cloud is marked by black squares; points contributing to the final regression line have a red dot inside of their square.

The Segmentation Algorithm. After stereo post-processing and ground manifold removal we segment the disparity map into consistent regions called *patches*: a patch is a 4-connected component such that any two pixels in this region can be connected by a 4-path inside the region such that the disparity difference between two consecutive disparity values is always less than a defined threshold. This segmentation rule is entirely defined in disparity space.

In other words, we decompose the disparity image into smooth stereo patches which can be adjacent but they are separated due to different disparity levels. Advantages of this segmentation method are that it is easy to implement, fast in execution, and results in a unique decomposition.

4 Correspondence Analysis and Tracking

Let $\mathcal{A}^t = \{P_0, P_1, ..., P_n\}$, $\mathcal{A}^{t+1} = \{Q_0, Q_1, ..., Q_m\}$ be the sets of patches obtained from two consecutive disparity images at times t and $t+1$. In the following, symbol P refers always to a patch extracted from frame f_t, and Q to a patch from frame f_{t+1}. We search for corresponding patches such that each P corresponds to one Q at most, and each Q to at most one P. Some P's or Q's may not have corresponding patches.

Dissimilarity Measure. For measuring the dissimilarity of two patches P and Q, both given as sets of pixels in the same ij-plane, we apply a metric [13] defined by the ratio of the cardinality of the symmetric difference of P and Q to the cardinality of their union:

$$\Gamma(P, Q) = \frac{card(P \cup Q) - card(P \cap Q)}{card(P \cup Q)} \tag{8}$$

This metric equals zero if and only if both sets are equal, and equals one if both sets have no pixel (i, j) in common.

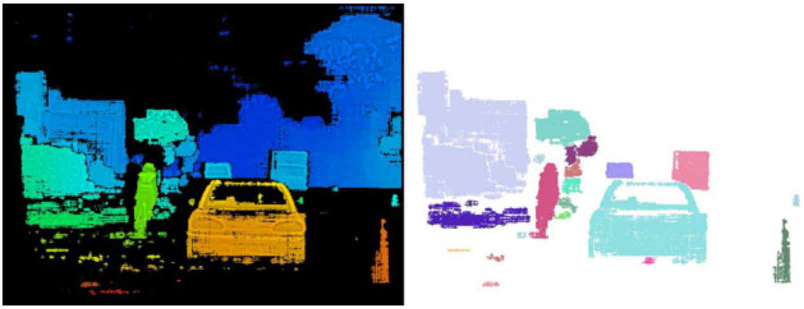


Fig. 5. Left: Result after mode filtering and ground manifold removal. Right: Result of the segmentation algorithm.

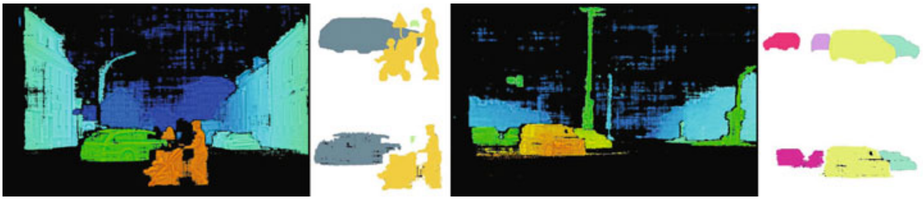


Fig. 6. Results of our segmentation experiment. Top: Silhouettes available as ground truth. Bottom: our segmentation results.

Two-Frame Correspondences. Because the position of patch P can change from frame f_t to frame f_{t+1} we use the relative motion as obtained by a dense optical flow algorithm (as specified above) to compensate for a translational component. Let $(u_p, v_p)^T$ be the flow at pixel p. We calculate the mean flow in the region occupied by P in f_t as

$$(u_P, v_P)^T = \frac{1}{card(P)} \sum_{p \in P} (u_p, v_p)^T \tag{9}$$

We shift the pixels in set P by $(u_P, v_P)^T$ into a new set $\overline{P}$.

For each $P_i \in \mathcal{A}^t$ we identify now the index j such that $\Gamma(\overline{P_i}, Q_j)$ is minimal for all $j = 0, ..., m$.

Temporal Tracking. In order to track corresponding stereo patches over multiple frames we apply the two-frame correspondence procedure repeatedly. As for all tracking algorithms, incorporating a-priori knowledge improves the robustness of the tracking results. Therefore, a very simple statistical filter is incorporated into our framework.

The filter stores for each tracked patch its history of τ corresponding patches at previous time slots. For experiments reported below we used $\tau = 6$, thus less than a quarter of a second.

For keeping the history of patches, we shift the positions of pixels in all patches of the history by the mean flow calculated from the current motion map. This means that the τ-th patch is shifted τ times when keeping track of the history.

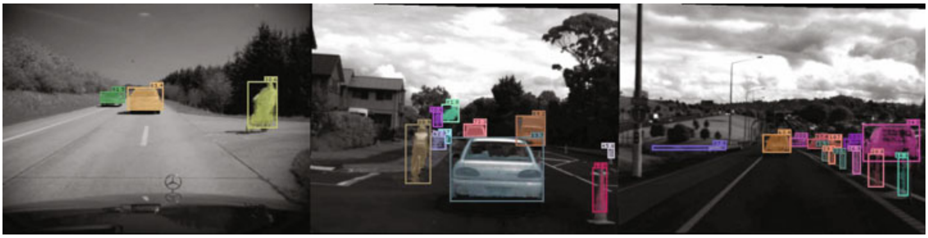


Fig. 7. Left to right: Bounding boxes of tracked patches for the sequences `intern on bike`, `cyclist` and `motorway`

To incorporate the prior knowledge of the history, a weight ω_t is assigned to each pixel of all patches based on the time instance t when it was added. The weights are then accumulated at each pixel location of the image domain over all patches of the current history. Thus we generate a new patch that represents the accumulated 'knowledge' of its history, and we use actually this generated patch for the two-frame correspondence.

Two-frame correspondence is now based on all pixels inside the $M \times N$ image domain where we have that $\omega_\Sigma > 0.5$ for the accumulated weight. Naturally, weights should decrease as t decreases, such that pixels from more recently added patches yield a higher contribution. For the six pixel sets in a history we use the weights (0.1, 0.1, 0.15, 0.15, 0.2, 0.3).

If there are patches $Q \in \mathcal{A}^{t+1}$ after the two-frame correspondence procedure that were not matched, then they are added as potentially new patches for the next time instance If these patches are matched in the next time instance then they are confirmed as being a new patch, otherwise they are discarded. A confirmed patch is considered to be identified if it is tracked at least over three frames. Its occupied area is defined by the accumulated weights from the history. If a patch is lost for more than τ frames then it is removed from the tracked object list.

5 Experiments and Results

We show that segmentation and tracking in the disparity domain is of adequate quality to support common DAS tasks. A comparison with other segmentation and tracking methods needs to be left to a future paper.

Used Data for Evaluation. We have chosen three image sequences from a public benchmark site [5] that provides experimental data for traffic-related applications. The first two sequences (from Set 4, see [12]) are called `cyclist` and `motorway` and consists of 400 frames each, the name of the third sequence (from Set 1, see [19]) is `intern on bike` and it has 250 frames. We run the proposed approach on those sequences and perform a visual evaluation. We are especially interested in stability of the segmentation and over how many frames an object is tracked. We focus our brief discussion on objects that are relevant for a traffic situation.

Segmentation Experiment. In [3] a method for segmentation and tracking of independently moving objects (IMO) is presented. The segmentation is based on evaluating probabilities at pixels whether they are in motion in real world coordinates or static, based on using scene flow and ego-motion information. Although this approach is not comparable to the approach presented here, we use the ground truth image provided by [3] (publicly available in Set 7 of [5]) as indication for the quality of the segmentation part of our algorithm. After incorporating ego-motion analysis into our framework (at a later stage), segmented objects can be labelled 'static' or 'in motion', as in a scene flow approach.

Results and Discussion. Figure 6 shows resulting stereo maps of two traffic scenes provided by the study [3] as generated by our SGM implementation with subsequent mode filtering and ground manifold removal. On the right of both maps, contours are shown that correspond to moving objects within the scene. Contours on the top are colour-coded ground truth segmentations. Below are results obtained with our segmentation procedure. Black pixels are caused by invalid disparities. The results indicate in general that prominent objects are segmented within reasonable DAS quality limits.

In the scene shown on the left of Fig. 6, the umbrella of the baby buggy is segmented as an individual object in our method. The reason is that fine structures like these are rarely reconstructed by stereo algorithms which results in disconnected smaller segments. A motion based segmentation is here of benefit, because umbrella and baby buggy should show a similar motion pattern.

In the scene shown on the right of Fig. 6, the two cars which are close to the ego-vehicle are clearly segmented. The two cars further away, however, are connected due to disparity noise. Again, a motion based approach should in theory cope with this situation by identifying different motion patterns. However, the immediately relevant objects closer to the ego-vehicle are segmented with very high accuracy.

Figure 7 shows Frame 141 from the `intern on bike` sequence, Frames 60 from the `cyclist` sequence, and Frame 20 from the `motorway` sequence. Bounding boxes enclose all pixels belonging to the tracked patches. The number in the top right corner indicates a mean estimate of the distance between ego-vehicle and object (in metres in the 3D world). Each object obtains an ID and a unique colour for its bounding box. If a tracked object is discarded and later picked up again, the colour will differ. For better visibility, we removed a few tracked objects in the visualization of results, but not in the tracking algorithm (e.g. all objects higher than 6 m and in cases of the `cyclist` sequence, everything left of the cyclist).

Reliable detection is possible up to 100 m when objects are in clear sight, meaning objects are lost only after a distance of 100 m or more, and are picked up by the tracking module before coming close to the 100 m mark. This range depends on camera parameters (e.g. baseline, focal length).

In the `cyclist` sequence we track cyclist, car, the van in the background, and the approaching truck robustly during the whole sequence. The truck is picked up at a distance of 137 m, and the van is lost at a distance of 130 m. The car is lost after 80 m, but this is mainly due because it vanishes at a corner behind a tree.

In the `intern on bike` sequence the program picked up all relevant objects until they disappear from the video. The bike is picked up at a distance of 89 m, the first approaching car at 121 m and the second approaching car at 95 m. In the motorway sequence all the numerous objects are tracked with a good robustness.

6 Conclusions

The main contribution of our work is to propose a concept and an implementation for solely disparity-based segmentation and tracking. Further contributions are the proposal of a mode filter for stereo post-processing which we found is crucial for an efficient flood flow segmentation algorithm. Or the application of a metric for set correspondences in the DAS context.

Methods and filters designed for 3D data analysis (see upper row in Fig. 1) could be applied to our method. Subsequent processing steps after disparity re-projection, such as 3D shape models or particle filters, will help to increase reliability. Another way of interpretation is, of course, that our presented approach can be considered as additional source of information for standard DAS methods.

Our results indicate the possibility of reliable segmentation and tracking in the disparity domain. It is difficult to compare our results with those provided by other methods, and more ground-truth data such as Set 7 on [5] would help.

Issues with our proposed segmentation approach: First, segmented objects may, depending on the stereo map, be connected in one frame and disconnected in the next; there is a need to generate a filter that resolves this behaviour based on prior knowledge. Second, small objects close to larger objects may not be segmented into independent objects because of insufficient disparity dynamics. Note that this applies in 3D data processing in general: Only if objects are moving then motion-based segmentation is able to identify different objects.

For the next stage we consider to include standard temporal filters to improve the stability and robustness such that more challenging traffic scenarios can also be processed with the same quality as for the sequences used in this paper.

Acknowledgments. The authors thank James Milburn for his implementation of the used dense optical flow algorithm.

References

1. Badino, H.: A Robust Approach for Ego-Motion Estimation Using a Mobile Stereo Platform. In: Jähne, B., Mester, R., Barth, E., Scharr, H. (eds.) IWCM 2004. LNCS, vol. 3417, pp. 198–208. Springer, Heidelberg (2007)
2. Badino, H., Franke, U., Pfeiffer, D.: The Stixel World - A Compact Medium Level Representation of the 3D-World. In: Denzler, J., Notni, G., Süße, H. (eds.) DAGM 2009. LNCS, vol. 5748, pp. 51–60. Springer, Heidelberg (2009)
3. Barth, A., Siegemund, J., Meißner, A., Franke, U., Förstner, W.: Probabilistic Multi-Class Scene Flow Segmentation for Traffic Scenes. In: Goesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) DAGM 2010. LNCS, vol. 6376, pp. 503–512. Springer, Heidelberg (2010)
4. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
5. enpeda. image sequences analysis test site,
http://www.mi.auckland.ac.nz/EISATS

6. Franke, U., Rabe, C., Badino, H., Gehrig, S.: 6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 216–223. Springer, Heidelberg (2005)
7. Gómez, G.D.: A global approach to vision-based pedestrian detection for advanced driver assistance systems. PhD thesis, Univ. Autónoma de Barcelona (2010)
8. Haller, I., Pantillie, C., Oniga, F., Nedevschi, S.: Real-time semi-global dense stereo solution with improved sub-pixel accuracy. In: IVS, pp. 369–376 (2010)
9. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. CVPR 2, 807–814 (2005)
10. Hirschmüller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. IEEE Trans. Pattern Analysis Machine Int. 31, 1582–1599 (2009)
11. Ivekovic, S., Clark, D.: Multi-Object Stereo Filtering in Disparity Space. In: COGIS (2009)
12. Klette, R., Krüger, N., Vaudrey, T., Pauwels, K., van Hulle, M., Morales, S., Kandil, F., Haeusler, R., Pugeault, N., Rabe, C., Markus, L.: Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database. IEEE Trans. Vehicular Technology (2011)
13. Klette, R., Rosenfeld, A.: Digital Geometry - Geometric Algorithms for Digital Picture Analysis. Morgan Kaufmann, San Francisco (2004)
14. Labayrade, R., Aubert, D., Tarel, J.-P.: Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In: IVS, pp. 646–651 (2002)
15. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IUW, pp. 121–130 (1981)
16. Oniga, F., Nedevschi, S., Meinecke, M.M.: Occupancy grids detected from dense stereo using an elevation map representation. In: WIT, pp. 133–138 (2009)
17. Petersson, L., Fletcher, L., Zelinsky, A., Barnes, N., Arnell, F.: Towards safer roads by integration of road scene monitoring and vehicle control. Int. J. Robotic Res. 25, 53–72 (2006)
18. Shimizu, M., Okutomi, M.: An analysis of subpixel estimation error on area-based image matching. In: Proc. Digital Signal Processing, vol. 2, pp. 1239–1242 (2002)
19. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In: IVCNZ, pp. 1–6 (2008)
20. Wedel, A., Badino, H., Rabe, C., Loose, H., Franke, U., Cremers, D.: B-spline modeling of road surfaces with an application to free space estimation. In: IVS, pp. 828–833 (2008)
21. Wedel, A., Meißner, A., Rabe, C., Franke, U., Cremers, D.: Detection and Segmentation of Independently Moving Objects from Dense Scene Flow. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 14–27. Springer, Heidelberg (2009)
22. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 739–751. Springer, Heidelberg (2008)
23. Wegener, P.: A technique for counting ones in a binary computer. Comm. ACM 3, 322 (1960)
24. Yu, Q., Araujo, H., Wang, H.: A Stereovision Method for Obstacle Detection and Tracking in Non-Flat Urban Environments. Autonomous Robots 19, 141–157 (2005)
25. Zabih, R., Woodfill, J.: Non-Parametric Local Transform for Computing Visual Correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 151–158. Springer, Heidelberg (1994)
26. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L^1 optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 214–223. Springer, Heidelberg (2007)
27. Zhao, L., Thorpe, C.: Stereo and neural network-based pedestrian detection. IEEE Trans. Int. Transportation Systems 1, 148–154 (2000)

Applications of Epsilon Radial Networks in Neuroimage Analyses

Nagesh Adluru[1,⋆], Moo K. Chung[1,2], Nicholas T. Lange[3],
Janet E. Lainhart[4], and Andrew L. Alexander[1]

[1] Waisman Center, University of Wisconsin-Madison, USA
adluru@wisc.edu
[2] Dept. of Brain and Cog. Sci., Seoul National University, Korea
[3] Dept. of Psychiatry and Biostatistics, Harvard University, USA
[4] Dept. of Psychiatry and Pediatrics, University of Utah, USA

Abstract. "*Is the brain 'wiring' different between groups of populations?*" is an increasingly important question with advances in diffusion MRI and abundance of network analytic tools. Recently, automatic, data-driven and computationally efficient framework for extracting brain networks using tractography and epsilon neighborhoods were proposed in the diffusion tensor imaging (DTI) literature [1]. In this paper we propose new extensions to that framework and show potential applications of such epsilon radial networks (ERN) in performing various types of neuroimage analyses. These extensions allow us to use ERNs not only to mine for topo-physical properties of the structural brain networks but also to perform classical region-of-interest (ROI) analyses in a very efficient way. Thus we demonstrate the use of ERNs as a novel image processing lens for statistical and machine learning based analyses. We demonstrate its application in an autism study for identifying topological and quantitative group differences, as well as performing classification. Finally, these views are not restricted to ERNs but can be effective for population studies using any computationally efficient network-extraction procedures.

Keywords: DTI, brain connectivity, tractography, brain networks, network measures, classification, toplogical group differences, autism.

1 Introduction

Population studies on brain connectivity networks are commonly performed using resting state functional magnetic resonance imaging (fMRI). These networks are called default mode networks (DMNs) and represent functional correlations between regions of the brain under rest [2]. These networks may not directly reflect the underlying structural organization of the brain white matter (WM). Diffusion tensor imaging (DTI) is a modality of MR imaging that is an exquisitely sensitive, non-invasive method to map and characterize the microstructural properties and macroscopic organization of the WM [3]. Streamline tractography methods

[⋆] Corresponding author.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 236–247, 2011.

on DTI data, albeit with limitations, are very useful for mapping major connections in the brain faithfully [4]. They have been used to develop in vivo dissection atlases [5] and build whole structural brain networks (e.g., Fig 1. of [6]). T1-weighted images are typically used for obtaining node regions for these networks. For example in [6], the cortex was parcellated into various regions using FreeSurfer[1] on a T1-weighted image. The main challenge in population studies using such brain networks is a DTI-T1 image co-registration since the problem of DTI to T1 co-registration is ill-posed and quite challenging: although there is contrast between white and grey matter in the T1-weighted images the contrast within white matter is not specific enough. More discussion on this can be found in Fig. 1 of the supplementary material[2]. This inter-modality image registration step forms a non-trivial hindrance for scalable studies of structural brain connectivity networks in population studies. Without a detailed evaluation study of such inter-modality registrations the connectivity analyses can be intricately confounded. Hence one of the key challenges in studying brain connectivity patterns in neuro-pathologies using DTI, short of the limitations of DTI, is *efficient* and *unbiased* designation of nodes and edges in the brain.

Recently a scalable framework that avoids the inter-modality registration has been proposed where, relying on well-validated tensor-based normalization methods, nodes are identified on the average DTI of a population using ϵ neighborhoods of end points of tracts obtained on the whole brain [1]. Some of the methods used *kd*-tree based search algorithms to identify the ϵ-radial nodes [7] while the others used a sequential elimination of tracts [1]. Except for the bias introduced from tractography, which is present in all streamline based methods, such a node generation does not introduce any bias from the ill-posed image registration processes. These methods are also computationally efficient: they can identify nodes and edges in a few seconds on a typical modern day computer [7].

The key extensions presented in this paper are: (1) We generate the nodes by first ordering the tracts by their length. Since the ϵ-neighborhood approaches depend on the sequence of tracts this is an important change as this removes the bias due to ordering of the tracts. (2) We enhance the edge properties by using geodesic information of the tracts and not just the count of the tracts. Such enhancements can result in increased sensitivity for statistical analyses. (3) Using the enhanced edge matrices we perform novel physio-topological as well as tract specific quantitative ROI analyses both in the setting of classical voxel based analyses (VBA) as well as classification.

2 Epsilon Radial Networks

Brain networks (BNs) are modeled similar to other network models that is as a collection of vertices (V) and edges (E). That is $\mathtt{BN} = \{\mathtt{V}, \mathtt{E}\}$. Tabel 1 summarizes

[1] http://surfer.nmr.mgh.harvard.edu

[2] http://brainimaging.waisman.wisc.edu/~adluru/ERN/supplementary.pdf

different modeling of the vertices and edges for contrasting with the epsilon radial networks (ERNs). In the default mode networks (DMNs) using resting fMRI, the vertices (node regions) are a function of blood oxygen level (BOLD) activations and the edges are based on temporal correlations between them. In the anatomical parcellation networks (APNs), the node regions are based on anatomical parcellation/segmentation [6]. In contrast, the nodes in ERNs are identified based on tracts themselves. This allows for identification of vertices (node regions) that have potential structural connectivity. Thus ERNs are completely DTI data-driven.

Table 1. Different models of brain networks

	DMNs	APNs	ERNs
V	f(BOLD activations)	f(segmentation $\pm$ registration)	f(tractography)
E	f(temporal correlations)	f(tractography)	f(tractography)

The ERNs are undirected and weighted networks and are constructed by adapting the framework and algorithms introduced in [7]. Briefly, the method uses the end points of the tracts to define the nodes by clustering neighboring tract end points into a set of spheres of ϵ radius which form the nodes for constructing connectivity matrices. Let $\mathtt{T}_{ij}$ denote the set of tracts connecting two vertices $i, j \in \mathtt{V}$. The original proposal defined $\mathtt{E} = \{|\mathtt{T}_{ij}|\}_{i,j \in \mathtt{V}}$. We propose that in addition to using tract counts as the edge strength, using the quantitative and physical properties using the geodesic pathway information of the tracts can enhance the ERNs.

That is we define $\mathtt{E} = \{|\mathtt{T}_{ij}|, \mathtt{quant}(\mathtt{T}_{ij}), \mathtt{physical}(\mathtt{T}_{ij})\}_{i,j \in \mathtt{V}}$. These enhanced ERNs can be more sensitive to group differences in population studies. In this paper we store the average fractional anisotropy (FA), mean diffusivity (MD) and axial diffusivity (AD) along tracts and the geodesic lengths of the tracts. Other diffusion based measures like radial diffusivity (RD), skewness, planarity, linearity and sphericalness may also be stored. In typical voxel based analyses an FWHM of 8mm smoothing is used to compensate for errors in spatial normalization. Hence we use an $\epsilon = 4$ to match the smoothing amount. The ERN nodes on the average template are shown in Fig. 1. As can be seen, the nodes have a good coverage of the brain regions and are generally in the grey/white matter boundaries as discussed in [7].

2.1 Properties of the ERNs

The nodes and edges of ERNs provide two fold advantages: (1) Provide an efficient way to extract various quantitative measures such as average FA, MD along the WM tracts and node regions. This is possible by extracting ROI masks using V and $\{\mathtt{T}_{ij}\}_{i,j \in \mathtt{V}}$. (2) Provide an efficient way to extract various topological properties of WM organization such as Rentian scaling, characteristic path length and clustering coefficient which are described next.

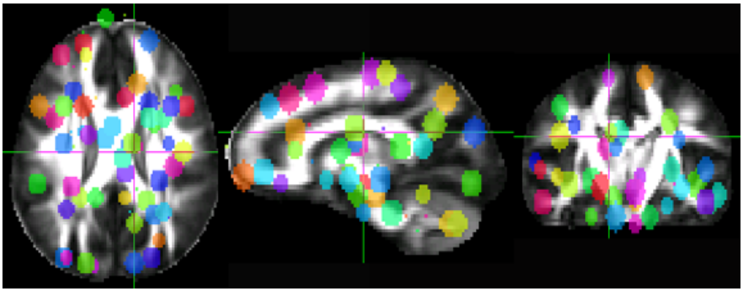


Fig. 1. The ϵ-radial nodes on the average DTI template are shown in random colors

(1) Rentian scaling: Imagine we can partition the vertices (`V`) of an `ERN` into n or physical partitions (e.g. cubes in a brain volume). Then it is likely that the following power law [8] holds for most of those partitions:

$$\mathcal{E} = k\mathcal{N}^r \tag{1}$$

where $\mathcal{E}$ is the number of connections crossing a partition and $\mathcal{N}$ is the number of nodes in that partition. k is called the Rent coefficient and $0 \leq r \leq 1$, the Rent exponent. When $k = 1$ and r is estimated using all the partitions from the $\log - \log$ relationships as:

$$\log(\mathcal{E}) = r \log(\mathcal{N}) \tag{2}$$

the estimated r is called Rentian scaling. If it is statistically significant for a given distribution of $\mathcal{E}$ and $\mathcal{N}$, that is the connections only scale linearly in the $\log - \log$ space, the network is considered efficient in terms of ”wiring cost” and physical embedding. Such features have been studied in the context of neuroimaging [9,10]. Following [8] the brain volume is partitioned into $n = 5000$ cubes in our experiments.

(2) Characteristic path length: The characteristic path length (`CPL`) of a network is defined as the average shortest path (`SP`) between all pairs of N vertices [11]:

$$\mathtt{CPL} = \frac{\sum_{(i,j)} \mathtt{SP}(i,j)}{N(N-1)} \tag{3}$$

It roughly indicates the efficiency of connectivity between regions in the network. The smaller the path length the more efficient the reachability is in a network. We would like to note the difference between this efficiency and the rentian scaling: the rentian scaling tries to characterize the efficiency in terms of resources needed to build the network while the characteristic path length tries to characterize the efficiency of the network in terms of connectivity/reachability and reflects “small worldness” of a network [11].

(3) Clustering coefficient: The clustering coefficient of a node ($\nu \in \mathtt{V}$) in a network is defined as the proportion of connections that it has to the rest of the network, i.e. the ratio of the number of edges connecting the node to the total number of *possible* edges that can connect the node [11]:

$$\texttt{CC}_\nu = \frac{|E_\nu|}{N(N-1)/2} \tag{4}$$

where $E_\nu = \{e_{\nu i}\}_{i \in \texttt{V} \setminus \nu}$ and $e_{\nu i}$ is the edge strength for e.g. in ERNs it would be $|\texttt{T}_{\nu i}|$. The clustering coefficient of an ERN is defined as the average clustering coefficient of a node in that network, i.e. $\texttt{CC}_{\texttt{ERN}} = (\sum_{\nu \in \texttt{V}} \texttt{CC}_\nu)/N$. The CC indicates the redundancy of connections in a network. Thus higher CC reflects the robustness of connectivity in a network. This is because the network can afford to lose some edges without losing connectivity to regions.

(4) Node-Strength: The strength of a node is a generalization of the degree of a node for weighted networks. It is defined as the sum of the weights of all edges connecting a node, i.e. $S_\nu = \sum_{i \in \texttt{V} \setminus \nu} e_{\nu i}$. The strength of a network can be defined as the average strength of all nodes in that network, i.e. $S_{\texttt{ERN}} = (\sum_{\nu \in \texttt{V}} S_\nu)/N$.

Thus ERNs are very useful in extracting different "views" of the DTI data for better sensitivity in neuroimage analyses. We use the implementations available in [12] to extract these measures on the ERN*s*.

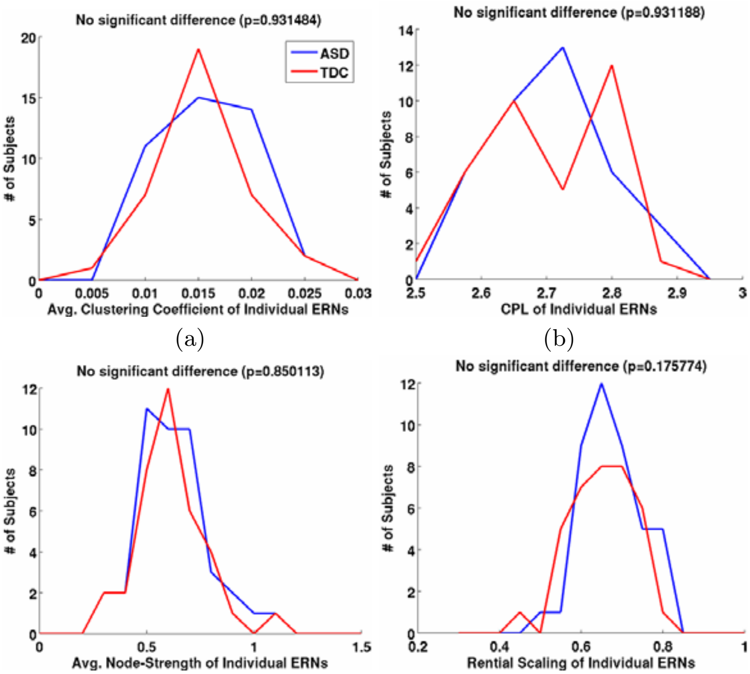
3 ERN Analyses in Autism

In this section we present various statistical analyses performed using different properties and measures extracted from ERNs. The details of the data and preprocessing can be found in the supplementary material [2]. First we look into three types of group differences: (1) Differences between average properties of the individual ERNs of the two groups. (2) Differences between the properties of the average ERNs of the two groups. (3) Differences between quantitative measures of the tissue extracted using individual ERNs, which involves $\texttt{quant}(\texttt{T}_{i,j})$. (4) Then using various features of the ERNs we perform classification using support vector machines [13]. (5) Finally we examine abnormal long vs. short range and hemispheric connectivity hypotheses in autism [14,15], which involves using $\texttt{physical}(\texttt{T}_{i,j})$.

(1) Differences between individual ERNs: The distribution of subjects in the two groups according to $\texttt{CC}_{\texttt{ERN}}$, $\texttt{CPL}_{\texttt{ERN}}$, $\texttt{S}_{\texttt{ERN}}$ and Rentian scaling are shown in Fig. 2. We can observe that there are no statistically significant differences between the two groups. This can be expected since the two groups are matched for age, IQ and handedness[3]. This also shows that our 'network-extraction process' does not introduce any bias into identifying group differences.

(2) Differences between the average ERNs: Let V denote the ϵ-radial nodes on the template and $\texttt{E}^i_{\texttt{ERN}}$ denote the edges of ERN of subject i. Then $\texttt{ASD}_{\texttt{ERN}} = \{\texttt{V}, \texttt{avg}(\texttt{E}^i_{\texttt{ERN}})_{i \in \texttt{ASD}}\}$ denotes the average ERN for the ASD group and $\texttt{TDC}_{\texttt{ERN}}$ can be

[3] Please see Fig. 2 of the supplementary material[2] for the matching information.



(a) (b)

(c) (d)

Fig. 2. The distribution of subjects in two groups according to different properties of their corresponding ERNs. (a) Average clustering coefficient, (b) Characteristic path length, (c) Average node-strength, (d) Rentian scaling. We can see that there is no statistically significant difference between the two groups in this sample-set using ERNs.

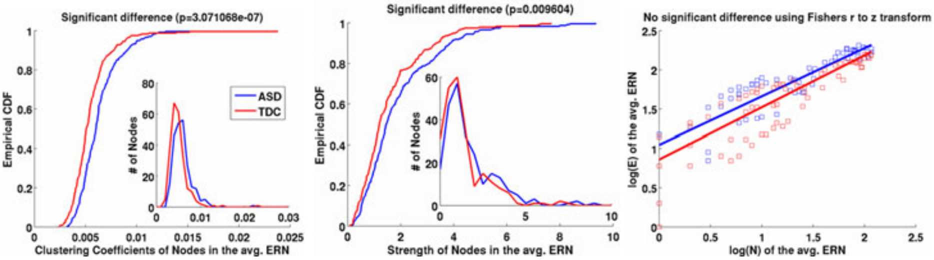


Fig. 3. Group differences using properties of the average ERNs. (a) Cumulative distribution function (CDF) of the nodes vs. clustering coefficient, (b) CDF of nodes vs. their strength. The significances of the differences are computed using Kolmogorov-Smirnov tests. (c) Rentian scaling with the corresponding log – log distribution of nodes in the partitions and their connections.

similarly defined. Fig. 3 shows the differences between the distributions of clustering coefficients of nodes, strengths of nodes and rentian scalings of $\texttt{ASD}_{\texttt{ERN}}$ and $\texttt{TDC}_{\texttt{ERN}}$. Since the distributions (showed in insets) are skewed we use Kolmogorov-Smirnov test [16], instead of two-sample t tests, to compare the significance of

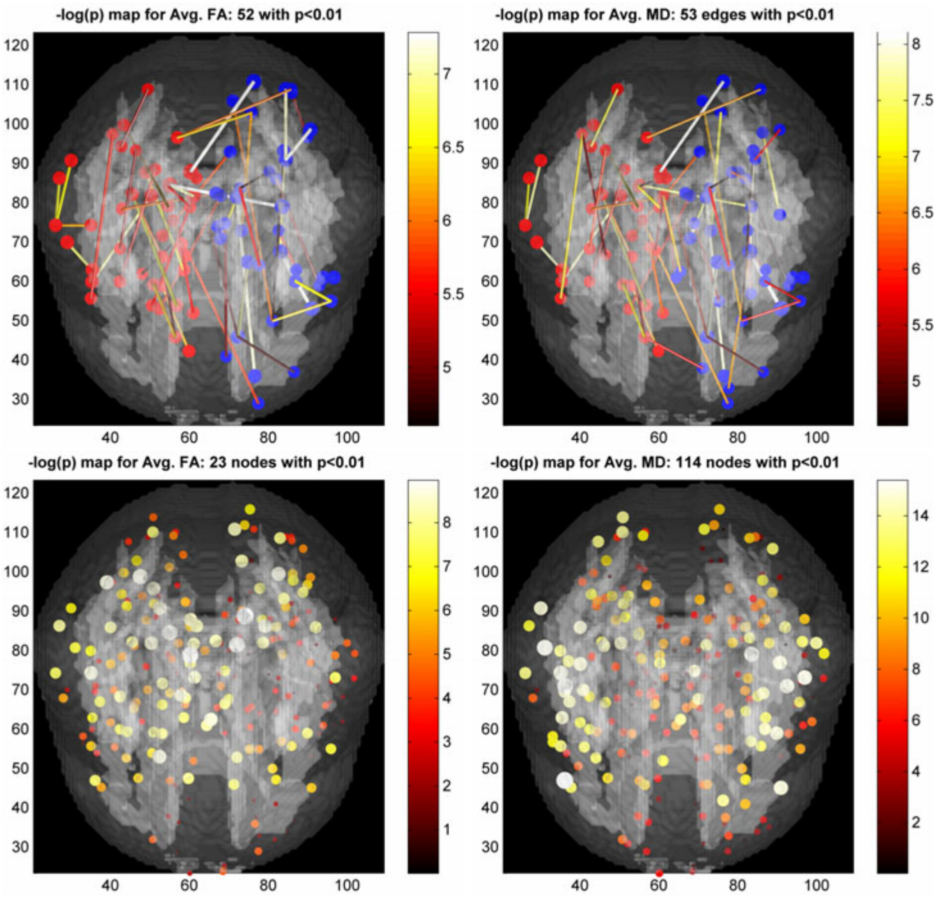

-log(p) map for Avg. FA: 52 with p<0.01
-log(p) map for Avg. MD: 53 edges with p<0.01
-log(p) map for Avg. FA: 23 nodes with p<0.01
-log(p) map for Avg. MD: 114 nodes with p<0.01
120
110
100
90
80
70
60
50
40
30
40
60
80
100
7
6.5
6
5.5
5
8
7.5
7
6.5
6
5.5
5
8
7
6
5
4
3
2
1
14
12
10
8
6
4
2

Fig. 4. Group differences using average quantitative measures (left - FA, right - MD) of the tissue masks obtained from V and $\{\mathtt{T}_{ij}\}_{i,j\in\mathtt{V}}$ in the individual ERNs. **Top row:** Significant edges. The nodes in the left and right hemispheres are colored red and blue respectively. **Bottom row:** Significant nodes. The size of the edges and nodes are proportional to the $-\log(p)$ values.

the differences between their corresponding cumulative distribution functions (CDFs). We can observe that there is decreased clustering coefficients and node strength in the ASD relative to the TDC. These two suggest under-connectivity of white matter in autism. There is no significant difference in the rentian scaling of the two average networks. This can also be expected as we do not expect a huge difference between the "wiring costs" of the brains of high-functioning ASD and TDC.

(3) Differences between quantitative measures: Here we perform classical ROI analyses using the masks obtained from V on the template and $\{\mathtt{T}_{ij}\}_{i,j\in\mathtt{V}}$ in the individual ERNs. The group differences using average FA and average MD

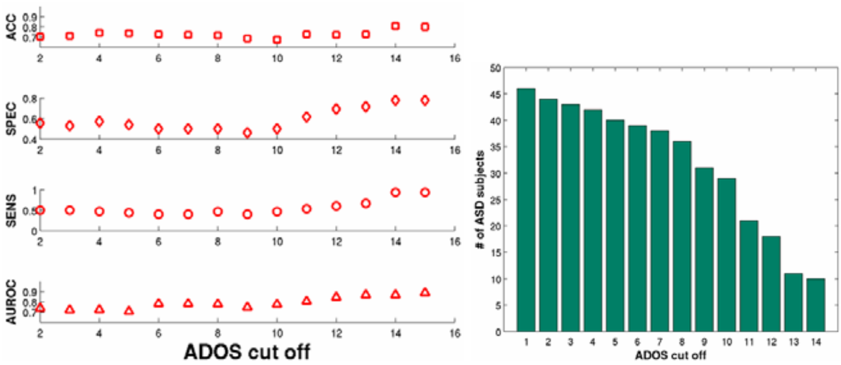
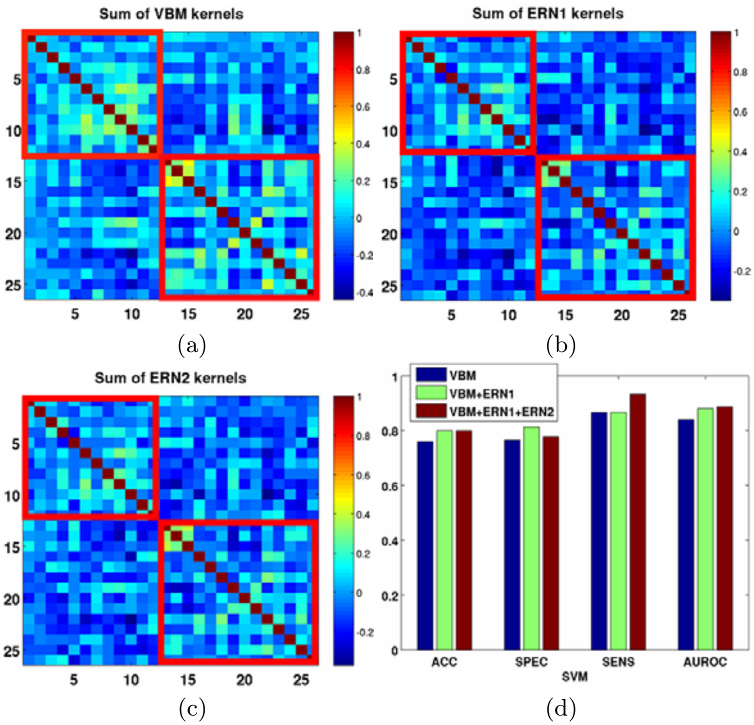


Fig. 5. Classification performance metrics as a function of the ADOS cut off used for the inclusion of ASD subjects. The ACC and AUROC are stable and peak at a cut off of 14. The other metrics show increase and saturate around 14. The right figure shows the change in the ASD sample size as the cut off increases.



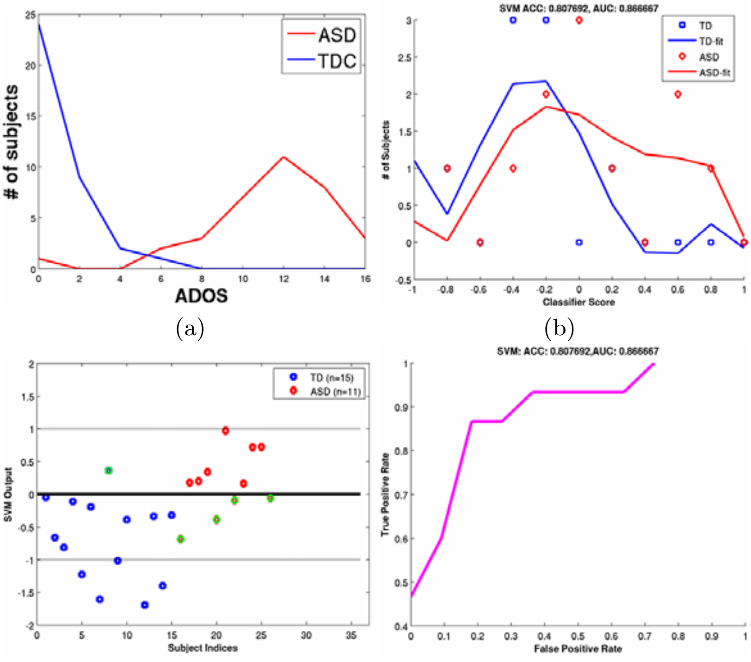
(a)

(b)

(c)

(d)

Fig. 6. Different kernels (features) and their effect on the SVM classification performance metrics. The highlighted red boxes show intra-class similarities for ASD (top-left) and TDC (bottom-right). In an ideal situation the similarities within the boxes should be higher than the similarities outside the boxes. The improvement in classification metrics due to addition ERN features is shown in (d).



(a)

(b)

(c)

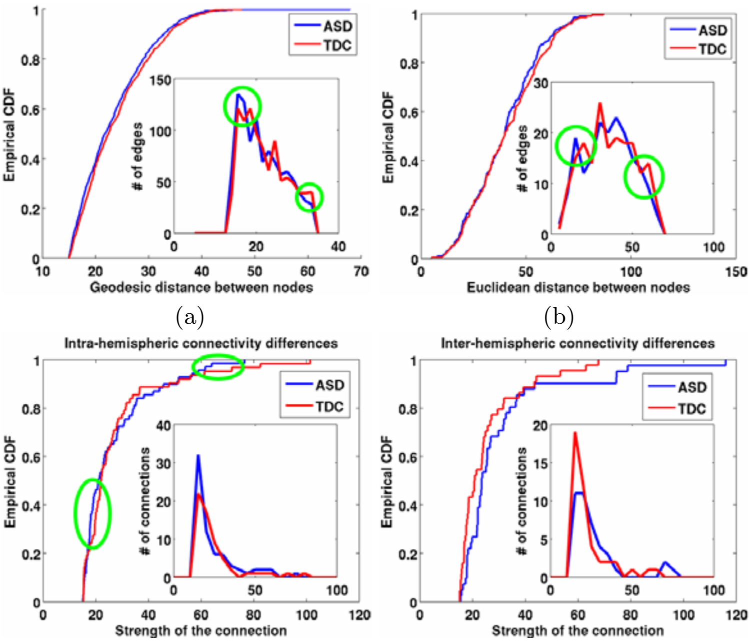
(d)

Fig. 7. (a) Distribution of subjects according to ADOS. (b) Distribution of subjects according to SVM output. In an ideal situation it should be as similar to (a) as possible. (c) SVM output for different subjects. The misclassified ones are encircled in green. (d) The ROC curve for the leave-one-out cross-validation.

in those masks are shown in Fig. 4. These differences can be attributed purely to the tissue property differences and are not confounded by network extraction procedure as shown by the failure to reject null-hypotheses using individual ERNs (Fig. 2). Thus using ERNs one can look into tissue differences by holding the topological properties constant when possible.

(4) Classification: Classification is a very challenging problem in autism studies especially using DTI. General leave-one-out cross-validation accuracies reported are in the high 70% to 80% [17,7,18]. An accuracy of 90% on an independently chosen test sample was reported in [19]. In this paper we report the performance of SVM classification using features extracted from ERNs (ERN1 and ERN2) as well as basic voxel based features of the WM (VBM).

- ERN1: *Average* FA, MD, AD on the node-regions in V.
- ERN2: FA, MD, AD at all the voxels in the mask obtained from all the node-regions in V.
- VBM: FA, MD, AD at all the voxels in the white matter mask on the template.



(a) (b)

(c) (d)

Fig. 8. Top row: Differences between long vs. short range connectivities using geodesic (a) as well as euclidean (b) distances between nodes. The empirical CDFs and the distributions of the edges (connections) are shown as insets. Although the differences are statistically not very significant ((a): $p = 0.0941$, (b): $p = 0.8723$), the encircled regions indicate support for the increased short-range and decreased long-range connectivities in ASD. **Bottom row:** Differences between intra and inter hemispheric connectivities between average ERNs of ASD and TDC. (c) Although the difference is not statistically significant ($p = 0.2443$), the encircled regions indicate support for the increased intra-hemispheric connectitvity for small and strong connections. (d) The inter-hemispheric connectivity is consistently lower for the ASD group ($p = 0.0624$) and is consistent with the finding in functional connecitvity [15].

For each of the above set of features we use both linear and radial-basis kernels for SVM classification. To measure the discriminative capacity of the features, we report classification performance metrics in the leave-one-out cross-validation setting, for different bootstraps of the data. The different metrics are accuracy (ACC), specificity (SPEC), sensitivity (SENS) and area under receiver operating characteristic (ROC) curve (AUROC). For the various bootstraps, we include all the TDC subjects with ADOS < 1 and include ASD subjects for different lower thresholds of ADOS as shown in Fig. 5. As the lower threshold of the ADOS increases the classification task becomes easier as the goal becomes separating extreme cases of ASD from TDC. Figs. 6 and 7 show the classification outputs for a particular bootstrap with TDC ($n = 15$) and ASD ($n = 11$) with ADOS $>$ 14 where the ACC and AUROC reach a maximum as shown in Fig. 5. Fig. 6 shows the sum of the kernels for VBM, ERN1 and ERN2 as well as the the improvement in classification metrics by the addition of ERN based features.

(5) Differences in long vs. short range and hemispheric connectivities: Such differences are one of the important hypotheses investigated in ASD. Indirect ways of characterizing these connectivities were proposed in the literature, e.g. using cortical thickness [20,21] and white matter volumes [14]. ERNs can provide a more direct way by looking at both the connectivities based on geodesic as well as euclidean distances between the node regions. Figs. 8 (a,b) show the group differences between these connectivities on $\mathtt{ASD}_{\mathtt{ERG}}$ and $\mathtt{TDC}_{\mathtt{ERG}}$. It has been indicated that ASD group has decreased inter-hemispheric *functional* connectivity [15]. ERNs can also be effectively used to investigate hemispheric *structural* connectivity differences, both intra and inter. Group differences between intra and inter hemispheric connectivities by plotting the distribution of the edges (connections) across different edge strengths are shown in Figs. 8 (c,d). We can observe decreased inter-hemispheric connectivity and increased intra-hemispheric connectivity in the ASD group. We would like to note that intra and inter hemispheric connections can also be thought of as a proxy to the short and long range connections respectively. To be sensitive to the changes, the same "easy" bootstrap sample (i.e. ADOS > 14 for ASD and ADOS < 1 for TDC) that was used for classification was also used for these two analyses.

4 Discussion

In this paper we extend recently proposed automatic, data-driven network extraction frameworks. These enhanced networks could potentially be more sensitive for network based analyses in population based neuroimaging studies. Such methods in addition to avoiding the bias of ill-posed inter-modality image registration (Fig. 1 of the supplementary material2) are computationally very efficient. However there are several limitations to be considered: (1) Tractography in the spatially normalized tensors needs to be validated against the tractography in the tensors native/acquired space. This is part of our on-going work. (2) The ϵ-radial nodes although cover important regions in the grey/white matter boundary, do not cover all possible regions of interest and can lead to false-negatives in group differences. Investigating potential extensions using techniques like Vietoris-Rips complex [22] are part of our future work. (3) The spatial normalization needed here may constrain the white matter topology to be too similar between subjects. The normalization causes the brain anatomy to have more consistent shape and size in the normalized space than they would in the native/acquired space. Hence, although the quantitative measures like FA, MD along the edges and node-regions might be preserved, this method may lose some sensitivity to individual differences of topology. Performing topological group differences without needing spatial normalization is also potentially an interesting line of work.

References

1. Chung, M., Adluru, N., Dalton, K., Alexander, A., Davidson, R.: Scalable brain network construction on white matter fibers. In: SPIE Medical Imaging (2011)
2. Greicius, M., et al.: Functional connectivity in the resting brain: a network analysis of the default mode hypothesis. Proc. Natl. Acad. Sci. 100, 253–258 (2003)
3. Jones, D., et al.: Non-invasive assessment of axonal fiber connectivity in the human brain via diffusion tensor MRI. Magn. Reson. Med. 42, 37–41 (1999)
4. Julien, D.J., Peled, S., Berezovskii, V., Delzescaux, T., et al.: Comparison of fiber tracts derived from in-vivo DTI tractography with 3D histological neural tract tracer reconstruction on a macaque brain. NeuroImage 37(2), 530–538 (2007)
5. Catani, M., Thiebaut de Schotten, M.: A diffusion tensor imaging tractography atlas for virtual in vivo dissections. Cortex 44(8), 1105–1132 (2008)
6. Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C., et al.: Mapping the structural core of human cerebral cortex. PLoS Biol. 6(7), e159
7. Adluru, N., et al.: Characterizing brain connectivity using ϵ-radial nodes: application for classifying autism. In: MICCAI Workshop on CDMRI (2010)
8. Danielle, S., et al.: Efficient physical embedding of topologically complex information processing networks in brains and computer circuits. PLoS Comp. Biol., 1–14 (2010)
9. Zhang, L., et al.: Quantifying degeneration of white matter in normal aging using fractal dimension. Neurobiol. of Aging 28, 1543–1555 (2007)
10. Chen, B., Hall, D., Chklovskii, D.: Wiring optimization can relate neuronal structure and function. Proc. Natl. Acad. Sci. 103, 4723–4728
11. Watts, D., Strogatz, S.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)
12. Rubinov, M., Sporns, O.: Complex network measures of brain connectivity: Uses and interpretations. NeuroImage 52, 1059–1069 (2010)
13. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001) Software, http://www.csie.ntu.edu.tw/~cjlin/libsvm
14. Jou, R., et al.: Reduced central white matter volume in autism: Implications for long-range connectivity. Psychiatry and Clinical Neurosci. 65, 98–101 (2011)
15. Anderson, J., et al.: Decreased interhemispheric functional connectivity in autism. Cerebral Cortex 21(5), 1134–1146 (2011)
16. Massey, F.J.: The kolmogorov-smirnov test for goodness of fit. J. Am. Stat. Assoc. 46, 68–78 (1951)
17. Adluru, N., et al.: Classification in DTI using shapes of white matter tracts. In: IEEE EMBS, pp. 2719–2722 (2009)
18. Ingalhalikar, M., Parker, D., Bloy, L., Roberts, T., Verma, R.: Diffusion based abnormality markers of pathology: Toward learned diagnostic prediction of ASD
19. Lange, N., et al.: Atypical diffusion tensor hemispheric asymmetry in autism. In: Autism Research, pp. 350–358
20. Herbert, M., et al.: Localization of white matter volume increase in autism and developmental language disorder. Ann. Neurol. 55, 530–540 (2004)
21. Hardan, A., Muddasani, S., Vemulapalli, M., et al.: An MRI study of increased cortical thickness in autism. Am. J. Psychiatry 163, 1290–1292 (2006)
22. Hausmann, J.: On the Vietoris-Rips complexes and a cohomology theory for metric spaces. Annals of Mathematics Studies 138, 175–188 (1995)

Road Image Segmentation and Recognition Using Hierarchical Bag-of-Textons Method

Yousun Kang[1], Koichiro Yamaguchi[2], Takashi Naito[2], and Yoshiki Ninomiya[2]

[1] Tokyo Polytechnic University
yskang@cs.t-kougei.ac.jp
[2] Toyota Central R&D Labs., Inc.
{yamaguchi,naito,ninomiya}@mosk.tytlabs.co.jp

Abstract. While the bag-of-words models are popular and powerful method for generic object recognition, they discard the context information for spatial layout. This paper presents a novel method for road image segmentation and recognition using a hierarchical bag-of-textons method. The histograms of extracted textons are concatenated to regions of interest with multi-scale regular grid windows. This method can learn automatically spatial layout and relative positions between objects in a road image. Experimental results show that the proposed hierarchical bag-of-textons method can effectively classify not only the texture-based objects, e.g. road, sky, sidewalk, building, but also shape-based objects, e.g. car, lane, of a road image comparing the conventional bag-of-textons methods for object recognition. In the future, the proposed system can combine with a road scene understanding system for vehicle environment perception.

Keywords: road image segmentation, hierarchical bag-of-textons, multi-scale.

1 Introduction

Intelligent Transport Systems (ITS) have developed significantly in the last few decades, and vehicle safety has been a particularly active research area [1]. The latest driving assistance systems include many vision-based applications such as lane detection, road detection, and pedestrian detection, which provide drivers with useful information [2]. Current vision-based intelligent vehicles are mostly focused on the detection of obstacles such as cars, bicyclists, and pedestrians.

However, an advanced driving assistance system may in the future be focused on *analysis* or *understanding* of a road scene than the detection of obstacles in road image. The scene understanding system requires integrated and/or advanced vision procedures, which are particularly relevant to image classification, object detection, and semantic segmentation. Among of them, semantic segmentation is a more complete image understanding system.

The role of semantic segmentation is central to visual interpretation and understanding to improve the effectiveness for vehicle environment perception. For example, by segmenting a road image, we can detect hazards or blind spots on the road. Such detection should consider the difference in potential risk for pedestrians standing on the

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 248–256, 2011.

road, on a crosswalk, or on the sidewalk. The probability of collision with a particular obstacle and the potential risk associated with a covert hazard can be estimated by segmenting a road image.

Therefore, automatically classifying pixels and parting meaningful regions in a road image is particularly helpful instance in vehicle safety field. This process is referred to as image labeling procedure, since its goal is to associate each pixel in the image with a label denoting a semantically meaningful part. In this paper, we investigate the problem of achieving recognition and segmentation of object classes in road image using hierarchical bag-of-textons method.

The bag-of-features method is one of the most popular and efficient for object recognition and image segmentation. It considers an object in an image as a set of unordered features extracted from local patches. The features are quantized into discrete visual words, with sets of all visual words referred to as a dictionary. Among various features, textons are representative dense visual words and they have been proven effective in categorizing materials as well as generic object classes [3-5]. In addition, textons are utilized in both object segmentation and recognition thanks to their high density and efficient [6].

However, the major drawback of the bag-of-features model is that it discards the spatial layout of visual words, which causes a serious problem for segmentation and recognition. In order to overcome the drawback, many researchers devote to develop the extension of the bag-of-feature model. Lazebnik *et al.* [7] proposed spatial pyramid matching (SPM) that utilizes the aggregated statistics of the local features on fixed sub-regions. SPM embeds a part of the spatial information over the whole image by partitioning an image into a sequence of sub-regions, so that they showed the good performance in scene categorization and object recognition.

In this paper, we propose a hierarchical bag-of-textons method that uses pairs of regular grid windows and neighborhood textons combined with multiple resolutions. Some objects in a road scene have a particular relation with other objects, e.g., cars are on the road, the road is below the sky, lanes surround the road, and so on. It is important to learn spatial layout and relative position between objects from the surrounding image. We uses a sequence of multi-scale grids and then computes a bag-of-textons histogram for each sub-region with different scale. Thus, the representation of the an object is the concatenation vector of all the histograms.

To classify the features of multi-class objects based on localized frequency of textons, we employ the Joint Boosting algorithm. We evaluate on our datasets including the variety road environment scenes and objects, e.g., road, tree, lane, sky, pole, sidewalk, car, and building. To assess how much a hierarchical bag-of-textons method helps with image segmentation and object recognition in road images, we have compared the recognition accuracy of conventional bag-of-textons methods. The experimental results show the proposed method improves the segmentation and recognition accuracy compared with the conventional bag-of-textons methods. As future work, we are interested in integrating the system into motion and semantic segmentation for vehicle environment perception. The proposed method can expand into a dynamic 3D scene analysis system in the near future.

The paper is organized as follows: Section 2 explain the filter bank and textonization process for road input image. Section 3 describes the feature extraction module for the hierarchical bag-of-textons method and the boosted classifier. Experimental results on performance and our conclusions are presented in the final two sections, respectively.

2 Textonization Process

In driving assistance system, infrared images are particularly useful for pedestrian detection of night vision systems and driver monitoring. The infrared is divided into near, far, and mid infrared, however, in this paper, we referred only to the near infrared. Near infrared is defined by water absorption, and the effect is formed by strongly reflecting off a person's exterior layer, and foliage, such as tree leaves and grass. Thus, in this paper, the near infrared and color image are available for road image segmentation and recognition. Input image has four bands consisting of a band of near infrared and three bands of color.

Convolving the four band image with a bank of linear spatial filters provides a good local descriptor of image patches and an effective statistical representation. Textons are typically a compact representation of filter bank responses for texture classification [9], image segmentation [10], and generic object recognition [11]. Kang *et al.* [8] compared the performance of various filter banks for the multiband image segmentation. Among the various filter banks, the 17-D set, which is proposed by Winn *et al.* [11], led to the best performance. The 17-D set consists of three Gaussians, four Laplacian of Gaussians (LoG), and four first-order derivatives of Gaussians. In order to implement the convolution of four bands image, we increase filter responses by adding the infrared intensity as a color intensity. We utilized the CIE Lab color space for three color bands. Fig. 1 shows how to expand the feature vectors of the 17-D set to 20-D set for a multiband image. The multiband images are convolved with a 20-D filter bank, and the cluster centers of the 20-D filter responses are utilized to generate image textons.

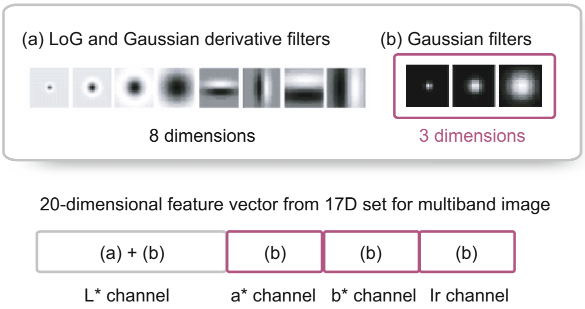


Fig. 1. The 20-D filter bank for a multiband image. The 17-D set filter banks are expanded to 20-D set for a multiband image.

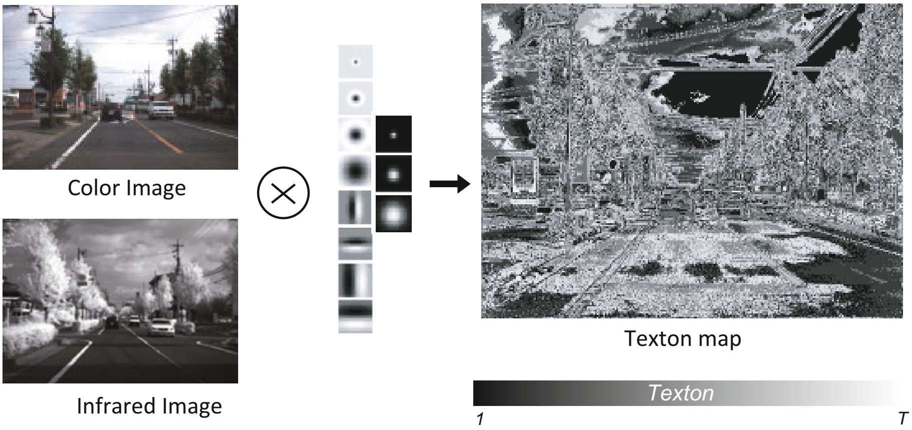


Fig. 2. Textonization image using 20D filter bank. Textons are represented by grayscale from 1 to T.

The road images are convolved with a 20-D filter bank and 20-D responses for all training pixels that are whitened to give zero mean and unit covariance. The K-means clustering is performed to quantize 20-D filter bank responses using a kd-tree algorithm [12]. We accomplished the textonization process using the code of *Calssification.NET* and *TextonBoost* [13] implemented by Shotton *et al.* [14]. Finally, each pixel in each image is textonized in the nearest cluster center, producing the texton map. Fig. 2 shows the texton map which is extracted from color and near-infrared image.

The filter responses are aggregated in the entire training set independently from class labels and clustered using K-means method to generate textons, which represent the visual words in a codebook of images. When a histogram of textons is created over a region of interest, we concatenate the histograms by using regular grids with multi-scale so as to learn automatically spatial relationship.

3 Hierarchical Bag-of-Textons

The bag-of-words models treat an object class as an unordered collection of visual words, sampling a representative set of image patches. However, it is important to extract the spatial configuration of an object and the contextual information from the surrounding image. It allows categorization and image segmentation algorithm to improve the performance by considering the context information of spatial layout. In road environment scene, there are spatial ordering constraints such as a car above a road and lanes are surrounded with road. It is necessary to order structural information between objects from the surrounding image. Therefore, we proposed a hierarchical bag-of-textons method using a spatial layout filter with multi-scale. The spatial layout filter with multi-scale is a pair (R, T) of a pyramid grid window R, and neighbor textons T. Our technique based on a bag-of-textons is capable of coping with spatial

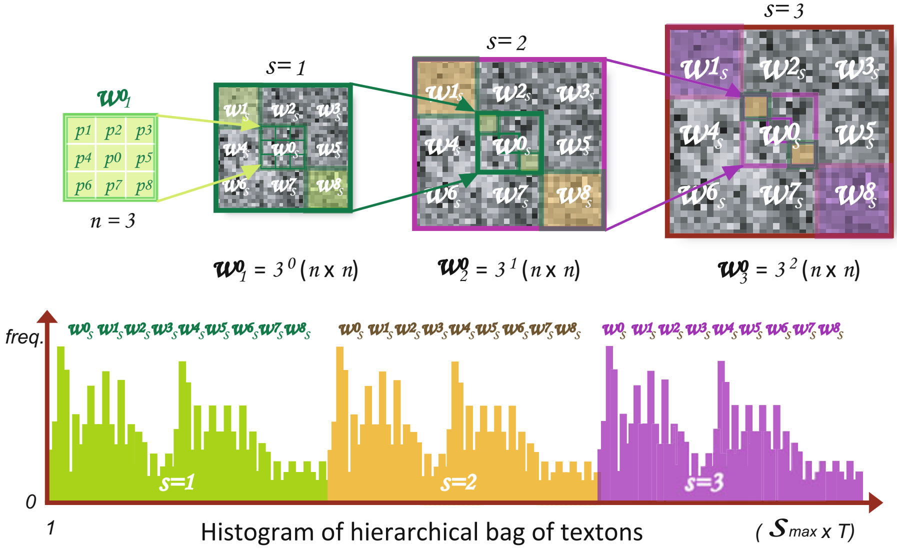


Fig. 3. The histogram of hierarchical bag of textons. Textons are represented by grayscale. The histogram of hierarchical bag of textons are normalized with window size.

ordering constraints of objects. The extracted features are sufficiently general to allow us to automatically learn the context informations for spatial layout and ordering constrain.

Fig. 3 illustrates how the hierarchical bag-of-textons are extracted to features using a multi-scale spatial layout filter. The original bag-of-textons method is computed over local rectangular regions from whole image. As illustrated in Fig. 3, the histogram of hierarchical bag-of-textons is extracted from grid windows increasing its resolution. At first, a set ω_s^0 of a candidate window with a center pixel p_0 are chosen as a $3^{s-1}(n \times n)$ window. The histogram of ω_s^0 concatenate from a top-left (ω_s^1) to bottom-right (ω_s^8) windows covering about $3^s(n \times n)$ the pixel area. The variable s indicates the step of multi-scale. At next, we increase scale step s to expand the features with multi-scale. The multi-scale windows method is effective combined with feature extraction module. We determined the scale step s from 1 to 3 and the initial window n to 3. At last, the size of multi-scale grid windows is normalized to generate a feature vector for object recognition.

A feature vector consists of the grid point's coordinates within the image as a location cue. We concatenated histogram consisting the multi-scale bag-of-texton to the feature vector. Outside the image boundary there is zero contribution to the feature response. We employ the Joint Boost algorithm [15] to select discriminative features of hierarchical bag-of-textons. Random feature selection and sub-sampling improve training time to generate several thousand weak learners. The learned strong classifier is an additive model of the form $H(c,i) = \sum_{m=1}^{M} h_m(c,i)$, summing the classification confidence of M weak classifiers. This confidence value can be reinterpreted as a

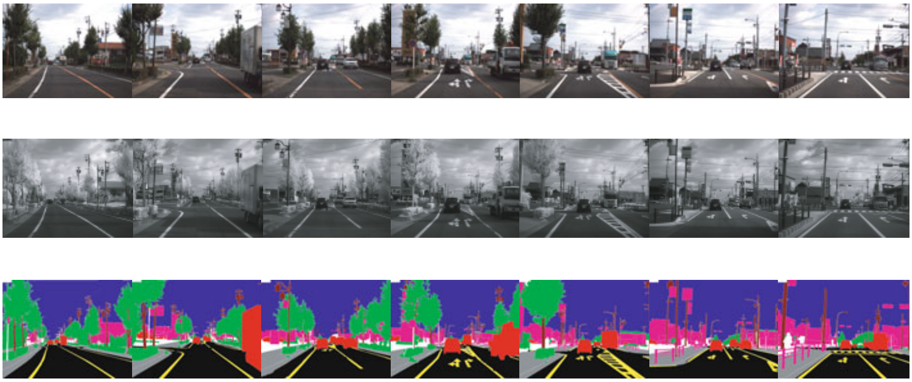


Fig. 4. Multiband Image Dataset Example training images: The first, second, third rows show color images, near-infrared images, and ground truth images, respectively. The assigned classes and colors were: road-black, lane-yellow, sky-blue, tree-green, car-red, trunk and pole-brown, sidewalk-gray, building-magenta, redundancy-white.

probability distribution using the soft-max[17] transformation to give the energy for optimal labeling. Thus, the confidence becomes:

$$P(c|x,i) = log \frac{exp\ H(c,i)}{\sum_c exp\ H(c,i)} \tag{1}$$

At last, the optimal labeling is found by applying the energy minimization algorithm based on the graph cuts [16]. The goal is to find a labeling f which minimizes some energy function. A standard form of the energy function is

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{p,q \in N} V_{p,q}(f_p, f_q) \tag{2}$$

where $N \subset P \times P$ is a neighborhood system on pixels. The $D_p(f_p)$ is a data function derived from the probabilities of Joint Boost assigning the label f_p to the pixel p. The $V_{p,q}(f_p, f_q)$ is a smoothness function that measures the cost of assigning the labels f_p, f_q to the adjacent pixels p, q :

$$V_{p,q}(f_p, f_q) = \begin{cases} C & \text{if } f_p \neq f_q \\ 0 & \text{if } f_p = f_q \end{cases} \tag{3}$$

where C is a constant.

4 Experimental Results

This section presents our experimental results for road scene labeling by using the proposed hierarchical bag-of-textons method. We investigated the performance of our system on road image datasets. Input images were captured using a prism-based multiband

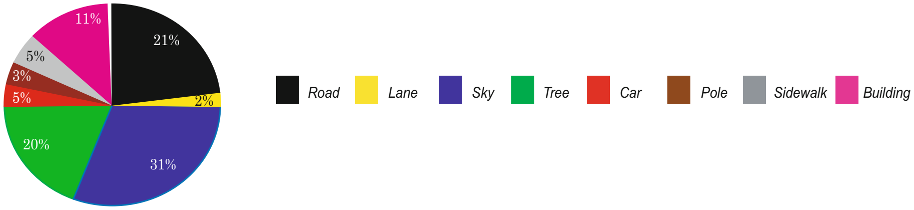


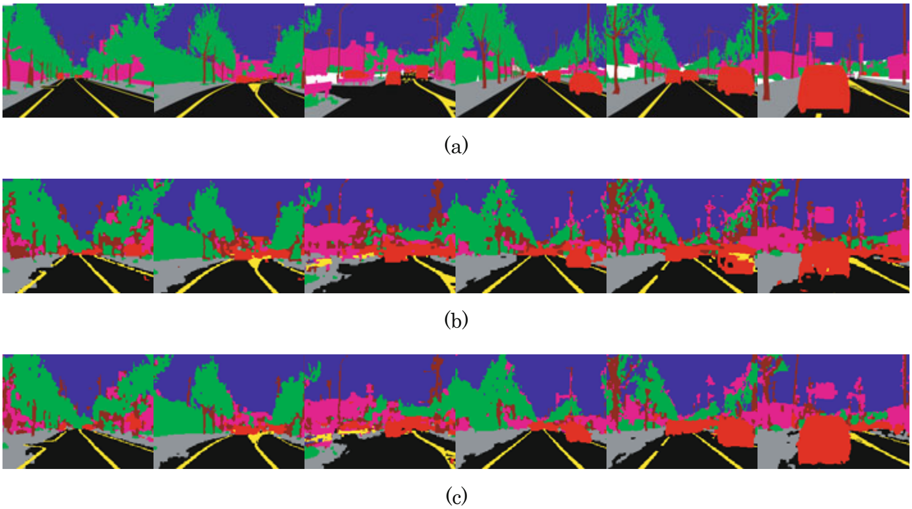
Fig. 5. The proportion of the training pixels in ground truth images

camera (JAI Inc., AD-080CL) mounted on a moving vehicle. The multiband camera can simultaneously obtain both images of color and near-infrared wavelengths. We proceeded to film 3 minutes of daytime footage and made labeled image for each sequence at one fps. In the case of video, each labeled frame could have potentially many other temporally related images associated with it. Each train and test set were captured from 90 video frames at 1.5 minutes. Our dataset contained 8 object classes and assigned a color as shown in Fig. 4. We extracted the features from ahead sequences to get the training patterns and the behind sequences were utilized for the test, which were not used in training image.

We compared the proposed method to conventional bag of texton method, which use single window size (15 × 15 pixels). The amount of training data is biased towards certain classes in our datasets so that we sampled the feature according to proportion of pixels of training set as shown in Fig. 5. We take training and test examples only at pixels lying on a 5 × 5 grid due to exhaustive memory and process time. However, the 20-D filter bank responses and texton map are calculated at full resolution (1024 × 768) for accurate pixel-wise segmentation. The texton number is $T = 297$ for train and test set. At boosting time, we have 10% random feature selection proportion with $M = 6000$ rounding. The constant C of the alpha-expansion algorithm of graph cuts is 0.3 for optimal labeling.

Fig. 6 shows example images for road scene labeling results. In Fig. 7, the table shows the overall recognition rate of the proposed method from total test images. Accuracy is computed by comparing the ground truth pixels to the inferred labeling. Segmentation performance is measured as both the category average accuracy (the average proportion of pixels correct in each category) and the global accuracy (total proportion of pixels correct). The category average is fairer and more rigorous, as it normalizes for category bias in the test set.

The average and the global segmentation accuracy of the proposed method are 80.6% and 84.6%, however, the average and the global of the conventional bag of textons method are 78.3% and 81.6%, respectively. Experimental results showed that the proposed method effectively segments road images and recognize objects in a road environment. The training and test datasets were real video sequences from a multiband camera mounted on a moving vehicle. However, we selected only daytime datasets for the experiments in this paper. If the lighting and weather conditions such as nighttime, snow, and rain are included, our system will struggle. Since robustness is essential for


(a)

(b)

(c)

Fig. 6. Experimental results of test images (a) Ground truth images (b) Labeling results of the conventional bag of textons method (c) Labeling results of the proposed method

Bag of Textons method	Road	Lane	Sky	Tree	Car	Pole	Sidewalk	Building	Average	Global
Conventinal BoT	89.7	91.1	90.6	74.6	85.6	54.3	83.9	57.2	78.3	81.6
Hierarchical BoT	93.5	93.1	92.7	77.6	86.8	53.9	84.5	62.9	80.6	84.6

Fig. 7. Total results in pixel-wise percentage accuracy on test sequences

ITS, we will attempt to integrate more reasonable features such as appearance features, motion and structural features, and lidar data. However, we have confirmed that the proposed system can play an important role in complex scene understanding for road environment perception. An optimized implementation of our system could be used as an advanced driving assist system.

5 Conclusion

This paper presented a new framework of semantic segmentation scheme for road environment perception using a hierarchical bag of textons method. Experimental results showed that the proposed method can be recognized more accurately than the conventional bag of textons method leading to a considerably better recognition of some objects such as road, tree, and building. Therefore, we can confirm that the proposed system is expected to play an important role in the complex road scene understanding system. In the future, by integrating the algorithm of shape-based objects recognition, we will use the proposed system to expand the road environment perception.

Acknowledgment. This work was supported in part by JSPS.

References

1. Bertozzi, M., Broggi, A., Fascioli, A.: Vision-based Intelligent Vehicles: state of the art and perspectives. Journal of Robotics and Autonomous Systems 32(1), 1–16 (2000)
2. Bertozzi, M., Broggi, A., Cellario, M., Fascioli, A., Lombardi, P., Porta, M.: Artificial vision in road vehicles. Proc. IEEE 90(7), 1258–1271 (2002)
3. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Proc. ECCV Int. Workshop on Statistical Learning in Computer Vision (2004)
4. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2005)
5. Winn, J., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2006)
6. Shotton, J., Johnson, M., Cipolla, R.: Semantic Texton Forests for Image Categorization and Segmentation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2008)
7. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2006)
8. Kang, Y., Kidono, K., Naito, T., Ninomiya, Y.: Multiband image segmentation and object recognition using texture filter banks. In: Proc. Int. Conf. on Pattern Recognition (2008)
9. Varma, M., Zisserman, A.: A Statistical Approach to Texture Classification from Single Images. Int. Journal of Computer Vision 62(1-2), 61–81 (2005)
10. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Int. Journal of Computer Vision 43(1), 7–27 (2001)
11. Winn, J., Criminisi, A., Minka, T.: Object Categorization by Learned Universal Visual Dictionary. In: Proc. IEEE Int. Conf. on Computer Vision (2005)
12. Beis, J.S., Lowe, D.G.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (1997)
13. Jamie Shotton's web site, `http://jamie.shotton.org/work/code.html`
14. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
15. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 19(5), 854–869 (2007)
16. Boykov, Y., Jolly, M.P.: Interactive Graph Cuts for optimal boundary and region segmentation of objects in N-D images. In: Proc. Int. Conf. on Computer Vision (2001)
17. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. Annals of Statistics 28(2), 337–407 (2000)
18. Kumar, S., Hebert, M.: A hierarchical field framework for unified context-based classification. In: Proc. of International Conference on Computer Vision (2005)

On the Security of a Hybrid SVD-DCT Watermarking Method Based on LPSNR

Huo-Chong Ling[1], Raphael C.-W. Phan[2], and Swee-Huay Heng[1]

[1] Research Group of Cryptography and Information Security,
Centre for Multimedia Security and Signal Processing,
Multimedia University, Malaysia
{hcling,shheng}@mmu.edu.my

[2] Loughborough University, LE11 3TU, United Kingdom
r.phan@lboro.ac.uk

Abstract. Watermarking schemes allow a cover image to be embedded with a watermark, for diverse applications including proof of ownership and covert communication. In this paper, we present attacks on watermarking scheme proposed by Huang and Guan. This scheme is hybrid singular value decomposition (SVD) based scheme in the sense that they employ both SVD and other techniques for watermark embedding and extraction. By attacks, we mean that we show how the designers' security claim, related to proof of ownership application can be invalidated. Our results are the first known attacks on this hybrid SVD-based watermarking scheme.

Keywords: singular value decomposition, watermarking, attacks, proof of ownership, ambiguity, discrete cosine transform.

1 Introduction

Nowadays, information is mostly stored in digital format. This results in widespread duplication of digital content and as a consequence, infringement of copyright has become an important issue that needs to be addressed. Digital watermarking has emerged as an efficient method to curb copyright protection issue. A digital watermarking scheme works by embedding the content owner's watermark into the content without significantly degrading the quality of the content. This watermark could be company's logo or any other text that identifies the owner. Once the case of copyright infringement is found, the owner takes the case of ownership claim to the authority, and proves ownership by performing the watermark extraction process on the claimed content to extract his watermark. Therefore, robustness of the watermarking scheme is an important factor, i.e. it should be infeasible for an attacker to remove, modify or prevent the extraction of an embedded watermark without visible distortions of the image.

In this paper, we concentrate on singular value decomposition(SVD)-based watermarking schemes. SVD is a linear algebra scheme that can be used for many applications, particularly in image compression [1], and subsequently for

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 257–266, 2011.

image watermarking [2–13]. For an N-by-N image matrix A with rank $r \leq N$, the SVD of A is defined as $A = USV^T = \sum_{i=1}^{r} u_i s_i v_i^T$ where S is an N-by-N diagonal matrix containing singular values (SVs) s_i satisfying $s_1 \geq s_2 \geq \ldots \geq s_r > s_{r+1} = \ldots = s_N = 0$, and U and V are N-by-N orthogonal matrices. V^T denotes the adjoint (transpose and conjugate) of the N-by-N matrix V. Since the SVs are arranged in decreasing order, the last terms will have the least affect on the overall image.

In past years, several SVD-based watermarking schemes [2–13] have been proposed. The most popularly cited scheme is due to Liu and Tan [12] that makes sole use of SVD for watermarking. They proposed to insert the watermark into the SVs of the cover image and demonstrated its high robustness against image distortion. However, Zhang and Li [14] and Rykaczewski [15] proved that the Liu-Tan scheme suffers from false-positive detection problem, i.e. the case where a watermarked image I_W^* does not contain a particular watermark W_A and yet it can be shown by an attacker that the watermarked image I_W^* does contain the watermark W_A. Therefore, the Liu-Tan scheme was not suitable to be used for proof of ownership application. In 2008, Mohammad et al. [13] proposed an improved variant of the Liu-Tan scheme and claimed that the improved version was able to solve the false-positive detection problem in the Liu-Tan scheme. However, their scheme was fundamentally flawed as proven by Ling et al. [16]. Other attacks on SVD-based watermarking schemes were found in [17–21].

In this paper, we furthermore show attacks on the hybrid SVD-based watermarking scheme proposed by Huang and Guan [9] that uses not just SVD but also discrete cosine transform (DCT) and local peak signal-to-noise ratio (LPSNR). By attacks, we mean that we show how the designers' security claim, related to proof of ownership application can be invalidated.

In Sect. 2, we recall the basics of the scheme proposed by Huang and Guan. We then present attacks on the scheme in Sect. 3 that invalidate the security claim of the designers. Experimental results verifying our attacks are given in Sect. 4, and Sect. 5 proposes countermeasure to the scheme. Finally, Sect. 6 concludes the paper.

2 Hybrid SVD-Based Watermarking Scheme

Huang and Guan [9] proposed a hybrid watermarking method that employs singular value decomposition (SVD), discrete cosine transform (DCT) and local peak signal-to-noise ratio (LPSNR). The SVD transform is performed on the watermark to get its singular values which are then embedded into selected DCT coefficients of the cover image based on Logistic mapping [22]. LPSNR is then applied to the watermarked image to exclude the block artifacts. The watermark embedding steps of the scheme are as follows:

E1. Denote cover image I as an N-by-N matrix and watermark W as an M-by-M matrix. I is divided into non-overlapping 8×8 sub-blocks I_k ($1 \leq k \leq \frac{N}{8} \times \frac{N}{8}$).

E2. Perform SVD on watermark W as:

$$W = USV^T. \tag{1}$$

E3. Select sub-blocks for watermark embedding using Logistic mapping [22], $X_{n+1} = \mu X_n$ (1 - X_n) which maps the unit interval into itself for $\mu \in$ [0, 4]. Select initial value $X_0 \in$ (0, 1) as the key and then drop the first 100 iterations to get a chaotic sequence

$$X_{101}, X_{102}, ..., X_{100+\frac{N}{8}\times\frac{N}{8}}. \tag{2}$$

where $\frac{N}{8} \times \frac{N}{8}$ is the length of the chaotic sequence (i.e. the number of 8 × 8 sub-blocks of cover image I).

E4. Construct another sequence $m_1, m_2, ..., m_{\frac{N}{8}\times\frac{N}{8}}$ from the sequence of (2) to index the sub-blocks in which the S in (1) is going to be embedded. If X_{100+i} is the jth bigger number in the sequence of (2), then $m_i = j$ where $(1 \leq i \leq \frac{N}{8} \times \frac{N}{8})$.

E5. Only sub-blocks with indices m_i are selected for embedding. DCT is performed on these sub-blocks as:

$$F^{m_i}(u, v) = DCT(I^{m_i}(r, c)). \tag{3}$$

where $1 \leq i \leq M$, $1 \leq u$, $v \leq 8$, $1 \leq r$, $c \leq 8$. $F^{m_i}(u,v)$ is the coefficient value at position (u,v) in DCT domain, whereas $I^{m_i}(r,c)$ is the coefficient value at position (r,c) in spatial domain.

E6. In each sub-block, one position (u_e,v_e) is selected for embedding S in (1) as:

$$F^{*m_i}(u_e, v_e) = F^{m_i}(u_e, v_e) + \alpha^{m_i} s_i. \tag{4}$$

where $(1 \leq i \leq M)$. Position (u_e,v_e) is chosen under the following rules:

- If s_i belongs to group A (which contains most energy of watermark), then (u_e,v_e) = (1,1).
- If s_i belongs to group B (which contains remaining energy of watermark), then (u_e,v_e) is chosen from set $C = \{(u, v) \mid 1 \leq u \leq 3, 1 \leq v \leq 3, u + v \leq 3\}$.

α^{m_i} is a scaling factor that determines the watermark strength, and it is determined by LPSNR value given in the following equation.

$$\text{LPSNR} = 10\ log_{10} \frac{(L-1)^2}{\frac{1}{8^2}\sum_{r=1}^{8}\sum_{c=1}^{8}[I^*_{m_i}(r,c) - I_{m_i}(r,c)]^2} \tag{5}$$

where L is the number of gray levels, $I^*_{m_i}(r,c)$ and $I_{m_i}(r,c)$ are the spatial coefficient values of the unwatermarked sub-block and the corresponding watermarked sub-block at the position (r,c), respectively.

E7. Perform inverse DCT on all the watermarked sub-blocks and substitute them for the corresponding sub-blocks in the cover image I to obtain the watermarked image I_W.

In order to perform the watermark extraction from the possibly distorted watermarked image I_W^*, the content owner needs to keep U and V (from Step E2), μ and X_0 (from Step E3) and α^{m_i} (from Step E6). The watermark extraction steps are as follows:

X1. Denote the possibly distorted watermarked image I_W^* as an N-by-N matrix. I_W^* is divided into non-overlapping 8×8 sub-blocks I_{Wk}^* $(1 \leq k \leq \frac{N}{8} \times \frac{N}{8})$.

X2. Repeat Steps E3 till E5 using μ and X_0 to find watermarked image's sub-blocks in which the SVs of watermark are embedded.

X3. Based on (4), the SVs of the watermark are extracted by:

$$s^*{}_i = \frac{(F^{*m_i}(u_e, v_e) - F^{m_i}(u_e, v_e))}{\alpha^{m_i}}. \tag{6}$$

The extracted sequence is described as $s^*{}_1$, $s^*{}_2$,...,$s^*{}_M$. $F^{*m_i}(u_e, v_e)$ and $F^{m_i}(u_e, v_e)$ are the DCT coefficient values of watermarked image's and cover image's sub-blocks at position (u_e, v_e) of index m_i respectively.

X4. The watermark is restored by:

$$W^* = US^*V^T. \tag{7}$$

where $S^* = \text{diag}(s^*{}_1,\ s^*{}_2,...,s^*{}_M)$.

Note that in the watermark embedding Step E2, the content owner needs to keep U and V so that he can use it later in the extraction Step X4.

2.1 On the Security Claim of the Huang-Guan Scheme

Huang and Guan claimed that their scheme was robust since the bigger singular values (SVs) which comprised most energy of the watermark were embedded into the DC components of the sub-blocks of the original cover image and LPSNR method was used. Therefore, their scheme was claimed to be usable in proof of ownership application. Nevertheless, in the next section, we present attacks on this scheme that violate the designers' claims.

3 Attacks on the Huang-Guan Scheme

We show in this section, how attacks can be mounted that invalidate the security claim made by Huang and Guan [9], namely that the scheme can be used for proof of ownership application. For the rest of the section, we will use Alice as the content owner and Bob as the attacker.

3.1 Attack 1

Our first attack invalidates the designers' claim that the Huang-Guan scheme can be used for proof of ownership application. We first recall the fact that in the embedding steps, Alice needs to keep the orthogonal matrices U and V of her watermark W, the parameters μ and X_0 and the scaling factor α^{mi} so that she can use it later in the extraction steps.

In order to launch the attack, Bob needs to obtain the watermarked image I_W^* and performs the embedding Steps E1 - E7 with I_W^*, his own watermark W_B, his own parameters μ_B and X_{B0} and his own scaling factor α^{Bmi} to obtain the watermarked image O. Both watermarked images I_W^* and O are perceptually correlated with each other since the same embedding steps are repeated. A dispute arises when Bob claims that he is the owner of O since he can extract his watermark W_B from O by supplying his own parameters μ_B and X_{B0}, and orthogonal matrices U_B and V_B of his watermark W_B. Alice could also lay equal claim to O since she too can extract her own watermark W from O by supplying her own parameters μ and X_0, and orthogonal matrices U and V of her watermark W. This leads to ambiguity because Bob lays equal claim as Alice, and therefore, no one can prove who the real owner of image O is.

This attack works because for an image I, its orthogonal matrices U and V due to SVD can preserve major information of the image [14, 15]. Therefore, if Bob uses his own U_B and V_B regardless of what the extracted singular matrix S^* is (as in (7)), he can still obtain a good estimate of the watermark W_B during the extraction process.

Besides that, the parameters μ and X_0 do not actually influence the robustness against this ambiguity attack. Their purpose is just to determine the cover image's sub-blocks that are used to embed the SVs of the watermark. Therefore, Bob can use his own parameters μ_B and X_{B0} to determine the sub-blocks that can be used to embed the SVs of his own watermark. Furthermore, Bob can use his own scaling factor α^{Bmi} to determine the strength of his embedded watermark in the watermarked image O.

This attack shows that the Huang-Guan scheme cannot be used for proof of ownership claim, directly invalidating the designers' claim that it can.

3.2 Attack 2

The second attack is another type of ambiguity attack described in Sect. 3.1. In this attack, an attacker can directly prove that the watermarked image I_W^* belongs to him also. The steps of our attack are as follows:

C1. Denote the possibly distorted watermarked image I_W^* as an N-by-N matrix and watermark W_B as an M-by-M matrix. I_W^* is divided into non-overlapping 8×8 sub-blocks I_{Wk}^* $(1 \leq k \leq \frac{N}{8} \times \frac{N}{8})$.

C2. Perform SVD on watermark W_B as:

$$W_B = U_B S_B V_B^T . \tag{8}$$

C3. Repeat Steps E3 - E6 using Bob's parameters μ_B and X_{B0}, his scaling factor α^{Bm_i} and his SVD components from Step C2. However, in Step E6, modify $F^{*m_i}(u_e, v_e)$ as follows:

$$F^{*m_i}(u_e, v_e) = F^{m_i}(u_e, v_e) - \alpha^{Bm_i} s_i. \tag{9}$$

The major change here is that the '+' operation in (4) is being replaced with the '−' operation.

C4. Perform inverse DCT on all the watermarked sub-blocks and substitute them for the corresponding sub-blocks in the image I_W^* to obtain the fake watermarked image O.

Now, instead of using O as the watermarked image, it is used as the cover image in the extraction process. The watermark extraction steps are as follows:

D1. Denote watermarked image I_W^* as an N-by-N matrix. I_W^* is divided into non-overlapping 8 × 8 sub-blocks I_{Wk}^* $(1 \leq k \leq \frac{N}{8} \times \frac{N}{8})$.

D2. Repeat Steps E3 - E5 using μ_B and X_{B0} to find I_W^*'s sub-blocks in which the SVs of watermark are embedded.

D3. The SVs of the watermark are extracted by:

$$s^*{}_i = \frac{(F^{*m_i}(u_e, v_e) - F^{m_i}(u_e, v_e))}{\alpha^{Bm_i}}. \tag{10}$$

The extracted sequence is described as $s^*{}_1$, $s^*{}_2$,...,$s^*{}_M$. $F^{*m_i}(u_e, v_e)$ and $F^{m_i}(u_e, v_e)$ are the DCT coefficient values of watermarked image I_W^*'s and cover image O's sub-blocks at position (u_e, v_e) of index m_i respectively.

D4. The watermark is restored by:

$$W_B^* = U_B S^* V_B^T. \tag{11}$$

where $S^* = \text{diag}(s^*{}_1,\ s^*{}_2,...,s^*{}_M)$.

In this attack, Bob uses the fake watermarked image O as the cover image, and proves that the watermarked image I_W^* belongs to him by extracting his watermark W_B from I_W^*. Alice, on the other hand, is also able to extract her watermark W from I_W^*. Therefore, a deadlock has resulted and no one can prove more than the others.

One may argue that Alice can also extract her watermark W from the fake watermarked image O because I_W^* is used as the cover image during the embedding steps and it contains Alice's watermark W. However, Bob can also extract his watermark W_B from Alice's original cover image I due to the properties of SVD [14, 15] as discussed in Sect. 3.1. Bob can just supply his watermark W_B and his fake watermarked image O to extract his watermark from Alice's original cover image I. This is illustrated in the experimental results section.

In either Attack 1 or Attack 2, Huang and Guan concentrated on ensuring that false negatives do not occur, i.e. the case where the watermarked image does indeed contain a watermark and yet it has been modified (while still maintaining perceptual similarity) such that the watermark can no longer be extracted.

Unfortunately, the designers did not treat the case of false positives, i.e. the case where the watermarked image does not contain a particular watermark and yet it can be shown by an attacker that the watermarked image does contain that particular watermark, which has never been embedded in the first place.

4 Experimental Results

In this section, we describe experiments that are carried out to further support our attacks in Sect. 3. Figure 1 shows a cover image, an owner's watermark, the watermarked image after going through the embedding Steps E1 - E7 and the extracted watermark, respectively. The values μ and X_0 used in Step E3 are 3.8 and 0.5 respectively.

Attack in Sect. 3.1 is carried out using the attacker's watermark in Fig. 2(a) and the watermarked image in Fig. 1(c). The values μ_B and X_{B0} used in Step E3 are 3.9 and 0.8 respectively. Figure 2(b) shows the watermarked image after the attack. The peak signal-to-noise ratio (PSNR) and the correlation coefficient (CC) between the watermarked image in Figs. 2(b) and 1(a) are 42.488 dB and 0.999 respectively. The closer the CC value is to either 1 or -1, the stronger the correlation between both images. This shows that both images are perceptually correlated, and the quality of the watermarked image after the attack is still in a very good condition. When no attack is introduced, the PSNR and the CC values between the cover image and the watermarked image are 45.389 dB and 0.999 respectively.

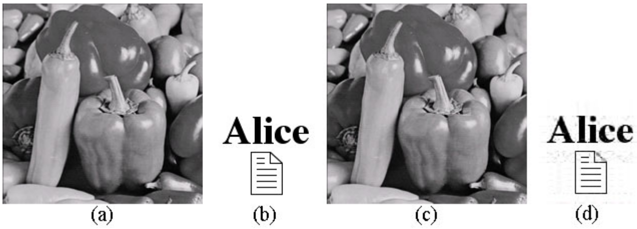


Fig. 1. (a)Original cover image with the size 200 × 200 (b)Owner's watermark with the size 100 × 100 (c)Watermarked image (d)Extracted watermark

Extraction process is then carried out on Fig. 2(b). Figures 2(c) and 2(d) show the extracted watermarks using the attacker's parameters and the owner's parameters respectively. Both attacker's watermark (PSNR = 25.665 dB, CC = 0.991) and owner's watermark (PSNR = 25.563 dB, CC = 0.985) can be extracted successfully.

Attack in Sect. 3.2 is then carried out, and Fig. 3(a) shows the watermarked image after the attack (PSNR = 42.488 dB, CC = 0.999). This watermarked image will then be used as the cover image in the extraction process. The result of the extraction process is that the attacker's watermark as shown in Fig. 3(b)

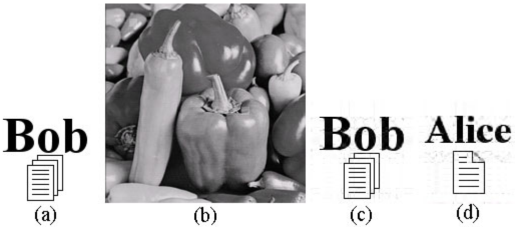


Fig. 2. (a)Attacker's watermark (b)Modified watermarked image after attack. (c)Extracted watermark using attacker's parameters (d)Extracted watermark using owner's parameters

(PSNR = 25.665 dB, CC = 0.991) can also be extracted out, besides the owner's watermark.

One may argue that Alice can also extract her watermark from Fig. 3(a) because the watermarked image in Fig. 1(c) is used as the cover image during the embedding steps and it contains Alice's watermark. However, Bob can also extract his watermark from Alice's original cover image in Fig. 1(a) due to the properties of SVD [14, 15]. In other words, Bob can just supply his watermark's U_B and V_B components and his modified watermarked image in Fig. 3(a) to extract his watermark from Alice's original cover image. This arguement is demonstrated and Figs. 3(c) and 3(d) show the results, where Alice's watermark and Bob's watermark can be extracted successfully.

Therefore, the Huang-Guan scheme is not suitable for protection of rightful ownership.

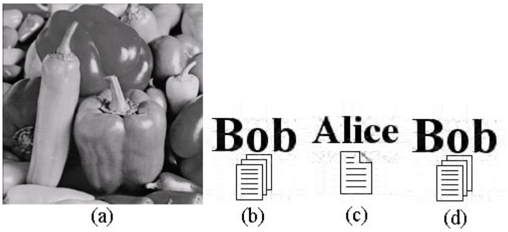


Fig. 3. (a)Modified watermarked image after the attack in Sect. 3.2 (b)Extracted watermark (c) Owner's extracted watermark from Fig. 3(a) (d) Attacker's extracted watermark from Fig. 1(a)

5 Countermeasure

One of the possible countermeasures is to embed the whole watermark into the DC coefficient of cover image's sub-blocks instead of using the SVs of the watermark. This will solve the dependant of orthogonal matrices U and V that influence the watermark being extracted from the watermarked image, at the expense of dropping the SVD in the embedding stage. Huang [23] has proposed a similar block-based watermarking scheme as in Huang and Guan scheme [9] using DCT and LPSNR. He suggested that other block-based transform domain, such as DFT, DWT and SVD can be used in the proposed scheme. However, as illustrated in the attacks in Sect. 3, it is not feasible to use SVD in the proposed scheme [23].

6 Conclusions

We have presented attacks on a watermarking scheme which is based on a hybrid use of SVD, DCT and LPSNR. These attacks work due to designers' oversight related to properties of the SVD, further supported by our experimental results. Huang and Guan [9] have neglected the fact that an image's orthogonal matrices U and V due to SVD can preserve major information of the image [14, 15]. Our attacks directly invalidate the security claim made by the scheme designers, namely use for proof of ownership application. Our results are the first known attacks on this scheme.

References

1. Andrews, H.C., Patterson, C.L.: Singular Value Decomposition(SVD) Image Coding. IEEE Trans. Commun. 24(4), 425–432 (1976)
2. Aslantas, V.: A Singular-Value Decomposition-Based Image Watermarking using Genetic Algorithm. AEU - Int. J. Electron. Commun. 62(5), 386–394 (2008)
3. Chang, C.C., Tsai, P., Lin, C.C.: SVD-Based Image Watermarking Scheme. Pattern Recognit. Lett. 26, 1577–1586 (2005)
4. Chang, C.C., Hu, Y.S., Lin, C.C.: A Digital Watermarking Scheme Based on Singular Value Decomposition. In: Chen, B., Paterson, M., Zhang, G. (eds.) ESCAPE 2007. LNCS, vol. 4614, pp. 82–93. Springer, Heidelberg (2007)
5. Chang, C.C., Lin, C.C., Hu, Y.S.: An SVD Oriented Watermark Embedding Scheme with High Qualities for the Restored Images. Int. J. Innov. Comput. Inf. Control. 3(2), 609–620 (2007)
6. Ganic, E., Eskicioglu, A.M.: Robust DWT-SVD Domain Image Watermarking: Embedding Data in All Frequencies. In: 2004 Workshop on Multimedia and Security, pp. 166–174. ACM, Magdeburg (2004)
7. Ganic, E., Eskicioglu, A.M.: Robust Embedding of Visual Watermarks using Discrete Wavelet Transform and Singular Value Decomposition. J. Electron. Imaging. 14(4), 43004 (2005)
8. Ghazy, R., El-Fishawy, N., Hadhoud, M., Dessouky, M., El-Samie, F.: An Efficient Block-by-Block SVD-Based Image Watermarking Scheme. Ubiquitous Comput. Commun. J. 2(5), 1–9 (2007)
9. Huang, F., Guan, Z.H.: A Hybrid SVD-DCT Watermarking Method Based on LPSNR. Pattern Recognit. Lett. 25, 1769–1775 (2004)
10. Lai, C.C.: A Digital Watermarking Scheme Based on Singular Value Decomposition and Tiny Genetic Algorithm. Digital Signal Processing 21(4), 522–527 (2011)
11. Lagzian, S., Soryani, M., Fathy, M.: A New Robust Watermarking Scheme Based on RDWT-SVD. Int. J. Intell. Inf. Process. 2(1), 22–29 (2011)

12. Liu, R., Tan, T.: An SVD-Based Watermarking Scheme for Protecting Rightful Ownership. IEEE Trans. Multimedia. 4(1), 121–128 (2002)
13. Mohammad, A.A., Alhaj, A., Shaltaf, S.: An Improved SVD-Based Watermarking Scheme for Protecting Rightful Ownership. Signal Processing 88, 2158–2180 (2008)
14. Zhang, X.P., Li, K.: Comments on "An SVD-Based Watermarking Scheme for Protecting Rightful Ownership". IEEE Trans. Multimedia. 7(2), 593–594 (2005)
15. Rykaczewski, R.: Comments on "An SVD-Based Watermarking Scheme for Protecting Rightful Ownership". IEEE Trans. Multimedia. 9(2), 421–423 (2007)
16. Ling, H.C., Phan, R.C.W., Heng, S.H.: Analysis on the Improved SVD-Based Watermarking Scheme. In: Kim, T.H., Adeli, H. (eds.) AST/UCMA/ISA/ACN 2010. LNCS, vol. 6059, pp. 143–149. Springer, Heidelberg (2010)
17. Ling, H.C., Phan, R.C.W., Heng, S.H.: Attacks on SVD-Based Watermarking Schemes. In: Yang, C.C., Chen, H., Chau, M., Chang, K., Lang, S.-D., Chen, P.S., Hsieh, R., Zeng, D., Wang, F.-Y., Carley, K.M., Mao, W., Zhan, J. (eds.) ISI Workshops 2008. LNCS, vol. 5075, pp. 83–91. Springer, Heidelberg (2008)
18. Ling, H.C., Heng, S.H., Goi, B.M.: Attacks on a Block Based SVD Watermarking Scheme. In: 6th Int. Conf. Inf. Technol.: New Generations (ITNG 2009), Las Vegas, Nevada, vol. 1-3, pp. 371–375 (2009)
19. Ting, G.C.W.: Ambiguity Attacks on the Ganic-Eskicioglu Robust DWT-SVD Image Watermarking Scheme. In: Won, D., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 378–388. Springer, Heidelberg (2006)
20. Xiao, L., Wei, Z., Ye, J.: Comments on "Robust Embedding of Visual Watermarks using Discrete Wavelet Transform and Singular Value Decomposition" and Theorectical Analysis. J. Electron. Imaging. 17(4), 40501 (2005)
21. Xing, Y., Tan, J.: Mistakes in the Paper Entitled "A Singular-Value Decomposition-Based Image Watermarking using Genetic Algorithm". AEU - Int. J. Electron. Commun. 64(1), 80–81 (2010)
22. Peitgen, H.O., Jurgens, H., Saupe, D.: Chaos and Fractals: New Frontiers of Science. Springer, New York (1992)
23. Huang, F.: A New General Transparency Model for Block-Based Watermarking Method. Int. J. of Network Secur. 7(2), 235–239 (2008)

Improved Entropy Coder in H.264/AVC for Lossless Residual Coding in the Spatial Domain

Jin Heo and Yo-Sung Ho

School of Information and Communicatitions
Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Republic of Korea
{jinheo,hoyo}@gist.ac.kr

Abstract. Since a block-based frequency transform is applied to residual data in lossy coding, it can reduce the spatial correlation efficiently. However, since residual data obtained from prediction is directly encoded without transform and quantization in lossless coding, there are some differences of the statistical properties in residuals between lossy and lossless coding. Based on the statistical characteristics of residuals in the spatial domain, we proposed an efficient context-based adaptive binary arithmetic coder (CABAC) for lossless residual coding. Experimental results show that the proposed CABAC provided approximately 19% bit saving, compared to the conventional CABAC.

Keywords: context-based adaptive binary arithmetic coding (CABAC), H.264/AVC, lossless coding, intra coding.

1 Introduction

Since H.264/AVC improves coding performance over previous video coding standards such as H.263 and MPEG-4 by using the state-of-the-art coding tools, it is currently the most powerful coding standard [1] [2]. Moreover, since H.264/AVC is known to provide high coding efficiency for lossy video coding, it has been used for a wide range of applications, including broadcast, storage, and video telephony.

Lossless compression has long been recognized as an important part for application areas that require high quality such as source distribution, digital cinema, and medical imaging. Recently, as the number of services for higher quality video representation is expanding, the importance for lossless coding is also increasing. However, since the majority of research pertaining to H.264/AVC has focused on lossy coding, it is not suitable for lossless coding.

In order to enhance coding performance for lossless coding, H.264/AVC included a transform-bypass lossless mode which uses prediction and entropy coding for encoding sample values, in the fidelity range extensions (FRExt) [3]. Although the new lossless mode of FRExt became a fairly efficient method for lossless video coder, it was not the best method for lossless coding. Therefore, more efficient coding techniques for prediction and entropy coding in lossless coding are still required.

Recently, new intra prediction methods, referred to as sample-wise *differential pulse-code modulation* (DPCM) [4] [5] were introduced for lossless intra coding.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 267–276, 2011.

However, since they still employed the conventional entropy coders, originally designed for discrete cosine transform (DCT) based lossy coding, new intra prediction methods have limited functionality.

The statistical characteristics of residual data are different between lossy and lossless coding; in lossy coding, residual data are the quantized transform coefficients in the frequency domain, whereas in lossless coding, residual data are just prediction residuals in the spatial domain without transform and quantization. Thus an improved context-based adaptive variable length coding (CAVLC) method was proposed for lossless coding [6]. In this paper, we proposed an improved context-based adaptive binary arithmetic coding (CABAC) method [7] for lossless intra coding based on the observed statistical properties of residual data in the spatial domain.

This paper is organized as follows. In Section 2, after we show an overview of H.264/AVC lossless coding, we present the CABAC encoder framework and the structure of CABAC for residual data coding. In Section 3, we explain our proposed CABAC method for lossless residual coding in the spatial domain. Experimental results are given in Section 4, and we draw our conclusion in Section 5.

2 Overview of H.264/AVC Lossless Coding and CABAC

2.1 H.264/AVC Lossless Coding

A typical encoding algorithm for lossy coding proceeds as follows. An input picture is split into macroblock and then each macroblock is encoded in intra or inter mode. The residual of the intra or inter prediction is transformed by a frequency transform. Finally the transform coefficients are quantized, entropy coded, and transmitted together with the side information. Fig. 1 shows the conventional H.264/AVC encoder structure.

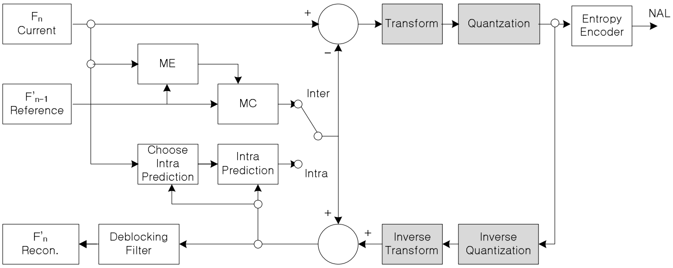


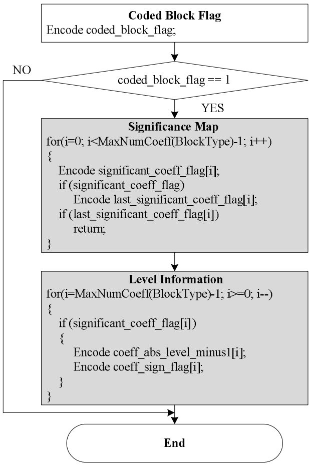
Fig. 1. Flowchart of CAVLC

However, an encoder structure for lossless coding is different from that for lossy coding. Since lossless data compression allows the exact original data to be reconstructed from the compressed bitstream, transform and quantization processes which cause a data loss are not included in the lossless encoder; the gray shaded stages in Fig. 1 are not included. Therefore, the residual obtained from prediction is directly encoded by the entropy coder. As a result, the residual data is handled only in the spatial domain during the entire lossless encoding process.

2.2 CABAC Framework and CABAC for Residual Data Coding

The encoding process of CABAC consists of four coding steps: binarization, context modeling, binary arithmetic coding, and probability update. In the first step, a given non-binary valued syntax element is uniquely mapped to a binary sequence; when the binary valued syntax element is given, the first step is bypassed. In the regular coding mode, each binary value of the binary sequence enters the context modeling stage, where a probability model is selected based on the previously encoded syntax elements. Then, the arithmetic coding engine encodes each binary value with its associated probability model. Finally, the selected context model is updated according to the actual coded binary value. Alternatively, in the bypass coding mode, each binary value is directly encoded via the bypass coding engine without using an explicitly assigned model.

Fig. 2 illustrates the CABAC encoding structure for a 4×4 sub-block of the quantized transform coefficients. First, the coded block flag is transmitted for each sub-block if the coded block pattern or the macroblock mode indicates that the specific sub-block has non-zero coefficients. If the coded block flag is zero, no further information is transmitted and the encoding process is terminated for the current sub-block; otherwise, the significance map and level information are sequentially encoded.



```
Encode coded_block_flag;
```

```
for(i=0; i<MaxNumCoeff(BlockType)-1; i++)
{
  Encode significant_coeff_flag[i];
  if (significant_coeff_flag)
      Encode last_significant_coeff_flag[i];
  if (last_significant_coeff_flag[i])
      return;
}
```

```
for(i=MaxNumCoeff(BlockType)-1; i>=0; i--)
{
  if (significant_coeff_flag[i])
  {
      Encode coeff_abs_level_minus1[i];
      Encode coeff_sign_flag[i];
  }
}
```

Fig. 2. Encoding structure of CABAC for residual data coding

Second, if *coded_block_flag* indicates that a sub-block has significant coefficients, a binary-valued significance map is encoded. For each coefficient, a 1-bit syntax element *significant_coeff_flag* is encoded in scanning order. If *significant_coeff_flag* is one, i.e., if a non-zero coefficient exists at this scanning position, a further 1-bit syntax element *last_significant_coeff_flag* is encoded. This syntax element states whether the current significant coefficient is the last coefficient or not.

After the encoded significance map determines the locations of all significant coefficients inside a sub-block, the values of the significant coefficients are encoded by using two syntax elements: *coeff_abs_level_minus1* and *coeff_sign_flag*. The syntax element *coeff_sign_flag* is encoded by a 1-bit symbol, whereas the *Unary/0th order Exponential Golomb* (UEG0) binarization method is used to encode the values of *coeff_abs_level_minus1* representing the absolute value of the level minus 1. The values of the significant coefficients are encoded in reverse scanning order.

3 Proposed CABAC Method for Lossless Coding

3.1 Analysis of the Statistical Properties of Residual Data in the Spatial Domain

In lossy coding, residual data represent the quantized transform coefficients in the frequency domain. The statistical characteristics of residual data in lossy coding are as follows. In a given sub-block, the probability of a non-zero coefficient existing is likely to decrease as the scanning position increases.

In lossless coding, however, residual data do not represent the quantized transform coefficients, but rather the differential pixel values between the original and predicted pixel values in the spatial domain. Therefore, the statistical characteristics of residual data in lossless coding are as follows. First, the probability of a non-zero differential pixel existing is independent of the scanning position, and the number of non-zero differential pixels is generally large, compared with the number of non-zero coefficients in the frequency domain.

Fig. 3 represents the probability distribution of non-zero residuals existing according to the scanning position. As expected, significant difference can be seen in the statistical characteristics between the residual data of the frequency (average value for the quantization parameter (QP) = 12, 24, and 36) and spatial domains.

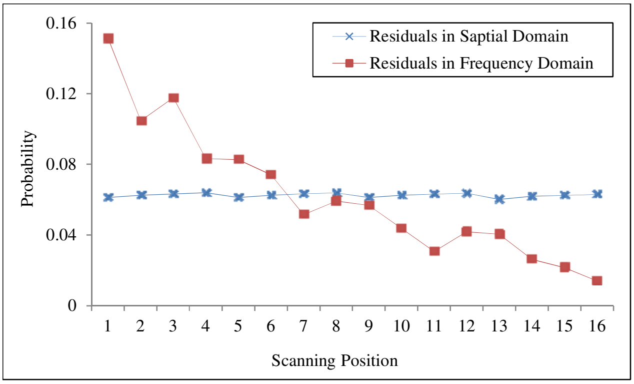


Fig. 3. Probability distribution of non-zero residuals according to the scanning position

Therefore, based on the above statistical characteristics of residual data in the spatial domain, we propose an efficient CABAC method for lossless coding in H.264/AVC by modifying the relevant coding parts of the conventional CABAC. In Fig. 2, the gray-shaded processes are modified in the proposed method for lossless coding. The coding procedure of the proposed CABAC can be summarized in the following steps.

Step 1: Encode whether the current sub-block contains non-zero differential pixel values (*coded_block_flag*).

Step 2: Encode whether the differential pixel value at each scanning position is non-zero up to the last scanning position (*significant_diff_pixel_flag*).

Step 3: Encode the absolute value of a differential pixel minus 1 with the modified binarization and context modeling methods (*abs_diff_pixel_minus1*).

Step 4: Encode the sign of a differential pixel (*diff_pixel_sign_flag*).

3.2 Significance Map Coding

In lossy coding, the occurrence probability of a non-zero coefficient is likely to decrease as the scanning position increases because residual data are the quantized transform coefficients, as shown in Fig. 3. Therefore, the significant coefficient tends to be located at earlier scanning positions. In this case, *last_significant_coeff_flag* plays an important role in the early termination of significance map coding.

However, in lossless coding, since neither transform nor quantization is performed, the occurrence probability of a non-zero differential pixel is independent of the scanning position, as shown in Fig. 3. Thus, the last non-zero differential pixel is located at the end of the scanning position. From the extensive experiments on the location of the last non-zero residual in a sub-block, we observed that the average position of the last non-zero residual data in lossy and lossless coding are approximately 14.71 and 7.75 (average value for the quantization parameter (QP) = 12, 24, and 36), respectively. In this case, it is meaningless to encode *last_significant_coeff_flag* to indicate the position of the last significant differential pixel. Therefore, we remove the *last_significant_coeff_flag* coding process and directly encode *significant_diff_pixel_flag*s at all scanning positions from 1 to 16 in the proposed significance map coding.

Scanning position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Transform coefficient level	8	-6	3	0	13	4	-9	1	0	11	-7	-2	5	-4	6	0
significant_coeff_flag	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	
last_significant_coeff_flag	0	0	0		0	0	0	0		0	0	0	0	0	1	

(a) Original method in lossy coding

Scanning position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Differential pixel value	8	-6	3	0	13	4	-9	1	0	11	-7	-2	5	-4	6	0
significant_diff_pixel_flag	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0

(b) Proposed method in lossless coding

Fig. 4. Example of significance map coding

3.3 Binarization for Differential Pixel Value

For the absolute value of the quantized transform coefficient (*abs_level*) in the frequency domain, the *Unary/kth order Exponential Golomb* (UEGk) binarization method is applied. The design of the UEGk binarization method is motivated by the fact that the unary code is the simplest prefix-free code in terms of implementation cost and it permits the fast adaptation of individual symbol probabilities in the subsequent context modeling stage [7]. These observations are only accurate for small *abs_level*s; however, for larger *abs_level*s, adaptive modeling has limited functionality. Therefore, these observations have led to the idea of concatenating an adapted truncated unary (TU) code as a prefix and a static Exp-Golomb code [8] as a suffix.

The UEGk binarization of *abs_level* has a cutoff value S = 14 for the TU prefix and the order k = 0 for the *Exponential Golomb* (EG0) suffix. Previously, Golomb codes have been proven to be optimal prefix-free codes for geometrically distributed sources [9]. Moreover, EG0 is the optimal code for a *probability density function* (pdf) as follows:

$$p(x) = 1/2 \cdot (x+1)^{-2} \;\; with \;\; x \geq 0 \tag{1}$$

The statistical properties of the absolute value of the differential pixel (*abs_diff_pixel*) in the spatial domain are quite different from those of *abs_level* in the frequency domain, as shown in Fig. 5. The statistical distribution of *abs_level* in the frequency domain is highly skewed on small values; however, in the spatial domain, the statistical distribution of *abs_diff_pixel* is quite wide. Moreover, in the figure, we can also observe that the TU code is a fairly good model for the statistical distribution of *abs_level* in the frequency domain; whereas, it is not appropriate for the statistical distribution of *abs_diff_pixel* in the spatial domain.

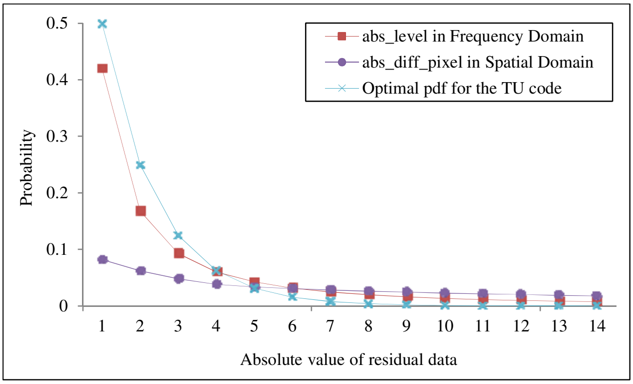


Fig. 5. Probability distribution of the absolute value and the optimal pdf of the TU code

In order to efficiently encode *abs_diff_pixel* in the spatial domain, we adjusted the cutoff value *S* of the TU prefix in UEG0 binarization. In Fig. 5, the optimal pdf curve for the TU code and the statistical distribution curve for *abs_diff_pixel* in the spatial domain intersect at an absolute value of 5. Moreover, as the absolute value increases, the statistical difference between the TU code and *abs_diff_pixel* becomes larger. Based on this observation, we determined a new cutoff value S = 5 for the TU prefix in the proposed binarization method.

In order to provide a good prefix-free code for lossless coding, we also determined an appropriate parameter k for the EGk code. The prefix of the EGk codeword consists of a unary code corresponding to the value $l(x) = \lfloor \log_2(x/2^k + 1) \rfloor$. The suffix is then computed as the binary representation of $x + 2^k(1 - 2^{l(x)})$ using $k + l(x)$ significant bits. Consequently, for EGk binarization, the number of symbols having the same code length of $k + 2l(x) + 1$ grows geometrically. Then, by inverting Shannon's relationship between the ideal code length and the symbol probability, we can find each pdf corresponding to an EGk having an optimal code according to a parameter k.

$$p_k(x) = 1/2^{k+1} \cdot (x/2^k + 1)^{-2} \ \ with\ x \geq 0 \tag{2}$$

where $p_k(x)$ is the optimal pdf corresponding to the EGk code for a parameter k. This implies that for an appropriately chosen parameter k, the EGk code represents a fairly good prefix-free code for tails of typically observed pdfs.

Fig. 6 presents the probability distribution of $p_k(x)$ for k = 0, 1, 2, and 3 and the occurrence probability distribution of *abs_diff_pixel* from 6 to 20, where *abs_diff_pixel*s up to 5 are specified by the TU code. In the figure, the probability distribution of $p_k(x)$ for k = 3 is well matched to the occurrence probability distribution of *abs_diff_pixel* in the spatial domain. This result implies that the EG3 code represents a fairly good approximation of the ideal prefix-free code for encoding *abs_diff_pixel* in the spatial domain.

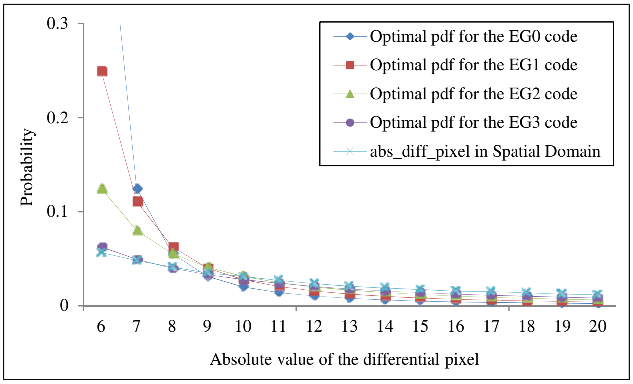


Fig. 6. Probability distribution of the optimal pdf of EGk code and *abs_diff_pixel*

Based on these observations, we designed an efficient binarization algorithm to encode *abs_diff_pixel* in the spatial domain. In the proposed algorithm, UEGk binarization of *abs_diff_pixel* is specified by new cutoff value S = 5 for the TU prefix and the order k = 3 for the EGk suffix.

4 Experimental Results and Analysis

In order to verify coding performance of the proposed method, we performed experiments on several test sequences for QCIF, CIF, and HD resolutions. We implemented our proposed method in the H.264/AVC reference software version JM 16.2 [10]. The encoding parameters for the reference software were as follows:

1) *ProfileIDC* = 244 (High 4:4:4)
2) *IntraPeriod* = 1 (only intra coding)
3) *QPISlice* = 0 (lossless)
4) *SymbolMode* = 1 (CABAC is used)
5) *ContextInitMethod* = 1 (adaptive)
6) *LosslessCoding* = 1 (lossless)

In order to evaluate coding performance of each proposed method, our experiment included two sections, based on the following settings:

1) *Method I*: modify significance map coding
2) *Method II*: *Method I* + modify binarization for differential pixel value coding

To verify efficiency of the proposed method, we performed two kinds of experiments. In the first experiment, six YUV420 format test sequences with QCIF, CIF, and HD resolutions were tested, as shown in Table 1. In the second experiment, we encoded only one frame (first frame) for each test sequence using our proposed method (*Method II*) and a well-known lossless coding technique, lossless joint photographic experts group (JPEG-LS) [11], used as a comparison for coding performance of our proposed method, as shown in Table 2.

Comparisons were made in terms of bit-rate percentage differences (Table 1) and compression ratio differences (Table 2) with respect to H.264/AVC CABAC and JPEG-LS, respectively. These changes were calculated as follows:

$$\Delta Saving\ \ Bits(\%) = \frac{Bitrate_{H.264/AVC} - Bitrate_{Method}}{Bitrate_{H.264/AVC}} \times 100 \tag{3}$$

$$Compression\ \ Ratio = \frac{Original\ \ image\ \ size}{Bitrate_{Method}} \tag{4}$$

In Table 1, we confirmed that the proposed method provided a better coding performance of approximately 18.9% bit saving via the QCIF, CIF, and HD resolution sequences, compared to the conventional CABAC. Table 2 presents the experimental results comparing JPEG-LS in terms of lossless intra coding, which again shows that the proposed method displayed the better coding performance in lossless coding.

In general, since video sequences contain more redundancy in time than in space, the accuracy of inter prediction is better than that of intra prediction. Thus, there are significant statistical differences between lossless intra and lossless inter coding. In other words, it is not easy to determine the best CABAC method that can generally be used for lossless video coding (for both intra and inter coding). Therefore, in this paper, we focused on the improvement of an appropriate CABAC for lossless intra coding.

Table 1. Comparison of the Performance Measures

Sequence	Image size (bits)	Method	Encoding bits (bits)	Saving bits (%)
Foreman (QCIF) 100 frames	30412800	H.264/AVC CABAC	14344176	0
		Method I	12857928	10.361
		Method II	12572368	12.352
Container (QCIF) 100 frames	30412800	H.264/AVC CABAC	14482016	0
		Method I	13034368	9.996
		Method II	11913968	17.733
Mobile (CIF) 100 frames	121651200	H.264/AVC CABAC	91371512	0
		Method I	85034984	6.935
		Method II	68152408	25.412
Tempete (CIF) 100 frames	121651200	H.264/AVC CABAC	79063136	0
		Method I	72756080	7.977
		Method II	60830560	23.061
Crowdrun (HD) 100 frames	2488320000	H.264/AVC CABAC	1250235376	0
		Method I	1120777696	10.355
		Method II	1047171240	16.242
Parkjoy (HD) 100 frames	2488320000	H.264/AVC CABAC	1283550664	0
		Method I	1155350512	9.988
		Method II	1043186200	18.727
Average		H.264/AVC CABAC		0
		Method I		9.269
		Method II		18.921

Table 2. Comparison of the Performance Measures

Sequence	Method	Compression ratio
Foreman (QCIF)	JPEG-LS	1.8179
	Method II	2.4190
Container (QCIF)	JPEG-LS	1.9030
	Method II	2.5527
Mobile (CIF)	JPEG-LS	1.4865
	Method II	1.7850
Tempete (CIF)	JPEG-LS	1.6556
	Method II	1.9998
Crowdrun (HD)	JPEG-LS	1.6802
	Method II	2.3762
Parkjoy (HD)	JPEG-LS	1.8664
	Method II	2.3853
Average	JPEG-LS	1.7349
	Method II	2.2530

5 Conclusions

In this paper, we proposed an improved context-based adaptive binary arithmetic coding (CABAC) method for H.264/AVC lossless residual coding in the spatial domain. Considering the statistical differences in residual data between the frequency and spatial domains, we modified the CABAC method based on the observed statistical characteristics of residual data in the spatial domain. Experimental results show that the proposed method provided an approximately 19% bit saving, compared to the H.264/AVC CABAC.

Acknowledgements. This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2011-(C1090-1111-0003)).

References

1. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Transactions on Circuits and Systems for Video Technology 13(7), 560–576 (2003)
2. Sullivan, G.J., Wiegand, T.: Video compression–from concepts to the H.264/AVC standard. Proc. IEEE 93(1), 18–31 (2005)
3. Sullivan, G.J., Topiwala, P., Luthra, A.: The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions. In: Proc. SPIE Conf., Special Session Adv. New Emerg. Standard: H.264/AVC, pp. 454–474 (2004)
4. Lee, Y.L., Han, K.H., Sullivan, G.J.: Improved lossless intra coding for H.264/MPEG-4 AVC. IEEE Transactions on Image Processing 15(9), 2610–2615 (2006)
5. Joint Video Team of ISO/IEC JTC1/SC29/WG11 and ITU-T Q.6/SG16, Lossless Intra Coding for Improved 4:4:4 Coding in H.264/MPEG-4 AVC, document JVT-P016.doc (2005)
6. Heo, J., Kim, S.H., Ho, Y.S.: Improved CAVLC for H.264/AVC lossless intra-coding. IEEE Transactions on Circuits and Systems for Video Technology 20(2), 213–222 (2010)
7. Marpe, D., Schwarz, H., Wiegand, T.: Context-based adaptive binary arithmetic coding in the H.264/AVC video compression. IEEE Transactions on Circuits and Systems for Video Technology 13(7), 620–636 (2003)
8. Teuhola, J.: A compression method for clustered bit-vectors. Information Processing Letters 7, 308–311 (1978)
9. Gallager, R.G., Van Voorhis, D.C.: Optimal source codes for geometrically distributed integer alphabets. IEEE Transactions on Information theory 21(2), 228–230 (1975)
10. Joint Video Team, Reference Software Version 16.2, `http://iphome/hhi.de/shehring/tml/download/old_jm/jm16.2.zip`
11. Weinberger, M.J., Seroussi, G., Sapiro, G.: The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. IEEE Transactions on Image Processing 9(8), 1309–1324 (2000)

Attention Prediction in Egocentric Video Using Motion and Visual Saliency

Kentaro Yamada[1], Yusuke Sugano[1], Takahiro Okabe[1],
Yoichi Sato[1], Akihiro Sugimoto[2], and Kazuo Hiraki[3]

[1] The University of Tokyo, Tokyo, Japan, 153-8505
{yamada,sugano,takahiro,ysato}@iis.u-tokyo.ac.jp
[2] National Institute of Informatics, Tokyo, Japan, 101-8430
sugimoto@nii.ac.jp
[3] The University of Tokyo, Tokyo, Japan, 153-8902
khiraki@idea.c.u-tokyo.ac.jp

Abstract. We propose a method of predicting human egocentric visual attention using bottom-up visual saliency and egomotion information. Computational models of visual saliency are often employed to predict human attention; however, its mechanism and effectiveness have not been fully explored in egocentric vision. The purpose of our framework is to compute attention maps from an egocentric video that can be used to infer a person's visual attention. In addition to a standard visual saliency model, two kinds of attention maps are computed based on a camera's rotation velocity and direction of movement. These rotation-based and translation-based attention maps are aggregated with a bottom-up saliency map to enhance the accuracy with which the person's gaze positions can be predicted. The efficiency of the proposed framework was examined in real environments by using a head-mounted gaze tracker, and we found that the egomotion-based attention maps contributed to accurately predicting human visual attention.

Keywords: Visual saliency, visual attention, first-person vision, camera motion estimation.

1 Introduction

Visual attention can be an important cue to infer the internal states of humans. Techniques to predict human visual attention have been employed in various applications in the area of, e.g., attentive user interfaces and interactive advertisements. One of the most direct ways of inferring visual attention is to measure the human gaze [7]; however, it is still a difficult task to measure our gaze in casual and unconstrained settings.

An alternative way of estimating the visual focus of attention is to use a visual saliency map model. Inspired by psychological studies on visual attention [24], Koch and Ullman proposed the concept of the saliency map model [17]. Itti et al. subsequently proposed a computational model [15] of visual saliency to identify image regions that attract more human attention. Following their study, many types of saliency map models have been proposed through the years [14,1,2,8,3,25]. Studies using gaze measurements [5,12,20] have also demonstrated that the saliency maps agree well with actual distributions of human attention.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 277–288, 2011.

Egocentric vision refers to a research field analyzing dynamic scenes seen from egocentric perspectives, e.g., taken from a head-mounted camera. Egocentric perspective cameras are suited for monitoring daily ego-activities, and hence accurate predictions of egocentric visual attention will be useful in various fields including health care, education, entertainment, and human-resource management. There has been much work on video attention analysis [18,21,13]; however, methods of analyzing egocentric visual attention have yet to be sufficiently explored. Saliency maps in these studies were computed from images shown to human subjects using monitors, and their effectiveness was evaluated against the gaze points given on the monitors. Hence, it still remains an unresolved question as to how we can predict visual attention accurately in egocentric videos that include visual motions caused by human head motion.

We propose a new framework in this paper to compute attention maps from egocentric videos using bottom-up visual saliency and egomotion information. Two kinds of egomotion-based attention maps, i.e., rotation-based and translation-based maps are computed in our framework and they are aggregated with the bottom-up saliency maps to produce accurate attention maps.

Camera motion has been employed to analyze attention in home videos [18,21]. Intentional human head motion in egocentric videos can have a stronger relationship with attention directed. Hillair et al. proposed a method of predicting egocentric visual attention in virtual reality environments based on the rotation factor of head movement [10,11]. Fukuchi et al. discussed the effect that focus of expansion (FOE) of moving pictures had in attracting human attention and they provided some experimental evaluations of FOE-enhanced saliency maps [6]. Although the basic idea behind our work was similar to that in these studies, we applied the framework to real egocentric scenes and motion-based maps were computed purely using input video without requiring additional sensors.

It is a well-known fact that humans tend to look at the center of images and a simple centering bias map can also contribute to enhancing the accuracy of saliency maps [16]. Our proposed attention maps can be seen as improved centering bias maps that are well-suited to egocentric vision. The effect of using motion-based attention maps is examined in a real setting using a mobile gaze tracker, and a comparison with a centering map is also discussed in Section 3.

2 Prediction of Visual Attention Using Saliency and Egomotion

The goal of this work was to predict visual attention by only using an egocentric video. Fig. 1 outlines the flow for our proposed framework. While bottom-up visual saliency maps are computed from input egocentric video, motion maps are computed using a person's egomotion. These additional motion maps are integrated into the visual saliency maps, and the resulting map achieves higher accuracy in predicting human attention. Details on the computations for the visual saliency maps and the motion maps are described in the following sections.

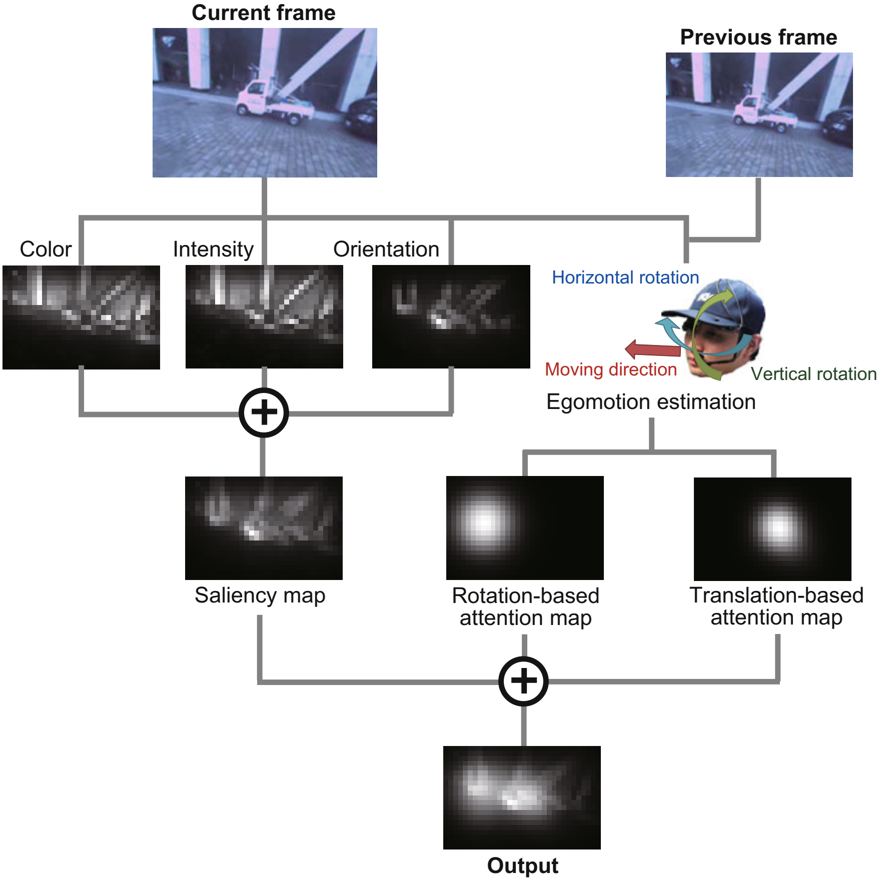


Fig. 1. Flow for our proposed framework. While bottom-up visual saliency maps are computed from input egocentric video, motion maps are computed using person's egomotion. These additional motion maps are integrated into visual saliency maps, and resulting map achieves higher accuracy in predicting human attention.

2.1 Computation of Visual Saliency Maps

We used the graph-based visual saliency (GBVS) model proposed by Harel et al. [8] in this work to compute the bottom-up saliency maps. Since it has previously been reported that saliency maps using dynamic features (motion and flicker) reduce the accuracy of saliency maps in egocentric scenes [26], we only employed static features, i.e., color, intensity and orientation to compute the saliency maps. As discussed above, the core concept in computational visual saliency is extracting regions with vastly different image features than their surrounding regions. Saliency maps in the GBVS model are generated by computing the equilibrium distributions of Markov chain graphs. Graphs are defined with nodes corresponding to pixels, and higher transition probabilities are assigned between dissimilar nodes (=pixels). Higher values are given in this way to nodes with distinctive image features in their equilibrium distribution and these can be used as saliency maps. Readers should refer to [8] for more details. Saliency maps are computed from each of the three features, and combined with equal weights to generate the final saliency map.

2.2 Computation of Attention Maps from Egomotion

Motion-based attention maps were computed using a person's egomotion in addition to the above visual saliency maps. We employed two kinds of attention maps in this work: rotation-based and translation-based. The computation consisted of three steps: 1) we estimated camera motion from the egocentric video, 2) estimated angular velocity and generating rotation-based attention maps, and 3) estimated the direction of movement and generated translation-based attention maps. We assumed that the camera's intrinsic parameters were known in this work and the lens distortion would be corrected through calibration. The camera was also assumed to be attached to the person's head so that its coordinates were identical to his/her visual field. Details on the three steps are described in what follows.

Estimation of Camera Motion. First, camera motion between two consecutive frames was computed using epipolar geometry, and rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ were obtained. Feature flows between the two frames were acquired using the Kanade-Lucas-Tomasi feature tracker [23,22], and an eight-point algorithm [9] was then applied to compute the fundamental matrix, $\boldsymbol{F}$. RANSAC [4] was used to robustly select the eight points without being affected by outliers caused by items such as moving objects. Since the intrinsic parameters were known, $\boldsymbol{R}$ and $\boldsymbol{t}$ could be obtained from $\boldsymbol{F}$.

Rotation-Based Attention Map. The rotation angle around each axis was computed from $\boldsymbol{R}$ in the second step, and the rotation-based attention map was generated using horizontal and vertical angular velocities. Let us denote the horizontal and vertical axes of the egocentric video as x and y, the camera's optical axis as z, and the rotation angles around these axes as $\theta_x, \theta_y, \theta_z$. Since it is assumed that the camera and the person's visual field share the same coordinates, the horizontal and vertical rotation angles of the head correspond to θ_y and θ_x. Given rotation matrix $\boldsymbol{R}$ and if we assume a x-y-z rotation order, θ_x and θ_y can be uniquely determined (θ_z is set to 0 if $\theta_y = \pm\frac{\pi}{2}$). By denoting the frame rate of the video as f [fps], the horizontal and vertical angular velocities can be written as $\omega_x = 180 f \theta_x / \pi$ and $\omega_y = 180 f \theta_y / \pi$.

We drew a 2-D Gaussian circle based on the angular velocities with a fixed variance to generate rotation-based attention maps. Hillair et al. [10] reported a strong correlation between gaze positions and angular velocities when the velocity was less than about $100[\deg/s]$. With larger velocity, Gaze positions tend to be almost fixed. According to their report, we define the center of the Gaussian (x, y) as illustrated in Fig. 2:

$$x = \begin{cases} \frac{\omega_y}{100} \cdot \frac{w}{k} & (|\omega_y| \leq 100) \\ \frac{w}{k} & (\omega_y > 100) \\ -\frac{w}{k} & (\omega_y < -100) \end{cases} \tag{1}$$

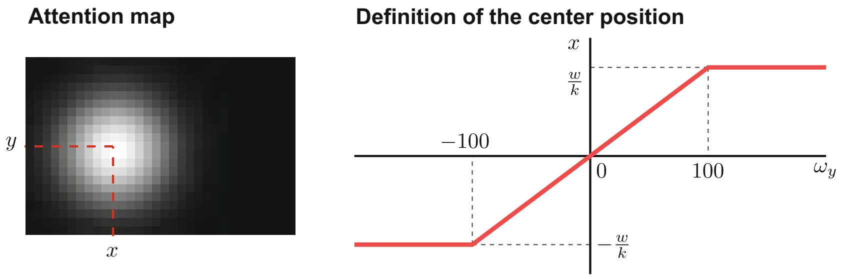


Fig. 2. Rotation-based attention map. 2-D Gaussian circle is drawn with fixed variance to generate rotation-based attention maps based on angular velocities. According to report by Hillair et al. [10], center of Gaussian is defined so that it is proportional to angular velocity within the range of $100[\deg/s]$.

and

$$y = \begin{cases} -\frac{\omega_x}{100} \cdot \frac{h}{l} & (|\omega_x| \le 100) \\ -\frac{h}{l} & (\omega_x > 100) \\ \frac{h}{l} & (\omega_x < -100), \end{cases} \tag{2}$$

where w, h indicate the width and height of the attention map and k, l are parameters according to the camera's angle of view.

Translation-Based Attention Map. Another attention map is generated in the third step based on the direction of the person's movement. The FOE of the input visual stimuli during translatory movements indicates the direction of movement. Similarly to [6], we generate the motion-based attention map based on the assumption that surrounding regions of the FOE attract more attention. We calculate the FOE of the input video as follows.

Egocentric videos usually contain independently moving objects and the person can also perform rotational movements. Therefore, the intersecting point of their feature flows does not always correspond to the FOE as illustrated in Fig. 3. We first rejected feature flows in this work that were identified as outliers when computing fundamental matrix $\boldsymbol{F}$ and only inlier flows were used in further processing.

Next, the rotational and translational components of the flow were separated. Let us denote the current image as $\mathrm{I}^{(t)}$ and the previous image as $\mathrm{I}^{(t-1)}$. If we can rotate $\mathrm{I}^{(t-1)}$ using the previously computed rotation matrix, $\boldsymbol{R}$, the relationship between the rotated image, $\mathrm{I}_R^{(t-1)}$, and I can be described by the translation vector, $\boldsymbol{t}$. If we denote the camera's intrinsic matrix as $\boldsymbol{A}$, pixel coordinates $\boldsymbol{m}^{(t-1)}$ and $\boldsymbol{m}_R^{(t-1)}$ of the feature point in $\mathrm{I}^{(t-1)}$ and $\mathrm{I}_R^{(t-1)}$ can be written in homogeneous coordinates as

$$\boldsymbol{m}^{(t-1)} \sim \boldsymbol{A}\boldsymbol{x}^{(t-1)}, \tag{3}$$

$$\boldsymbol{m}_R^{(t-1)} \sim \boldsymbol{A}\boldsymbol{x}_R^{(t-1)},, \tag{4}$$

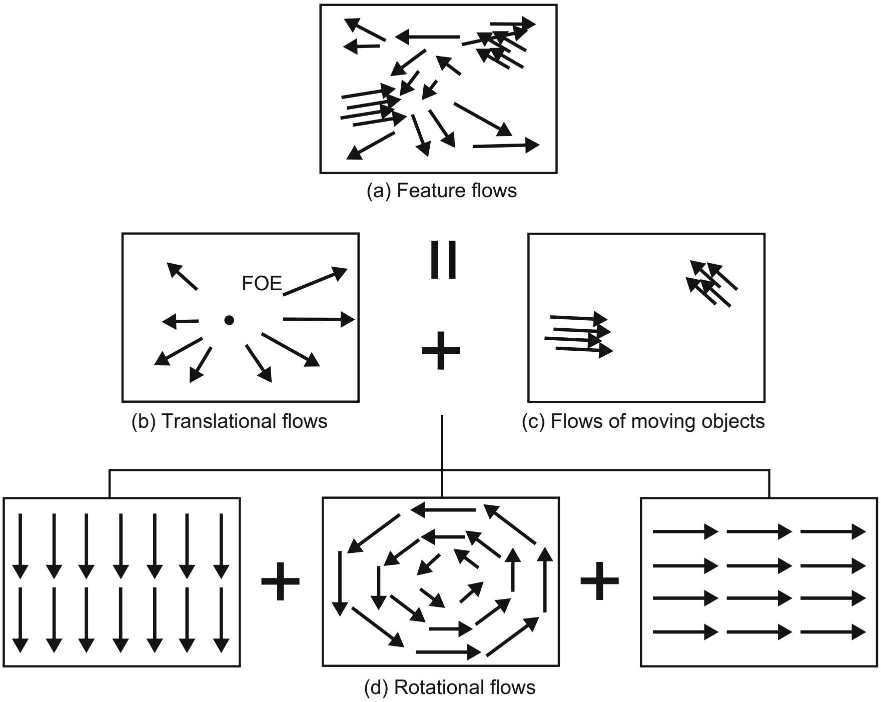


Fig. 3. Components of feature flows. Intersection point of translational flow (b) corresponds to Focus of Expansion (FOE) and it indicates direction of camera movement. However, feature flows computed from egocentric video (a) include flows caused by independently moving objects (c) and rotational movements (d). Components corresponding to (c) and (d) must first be separated from computed flow (a) to estimate FOE of input frame.

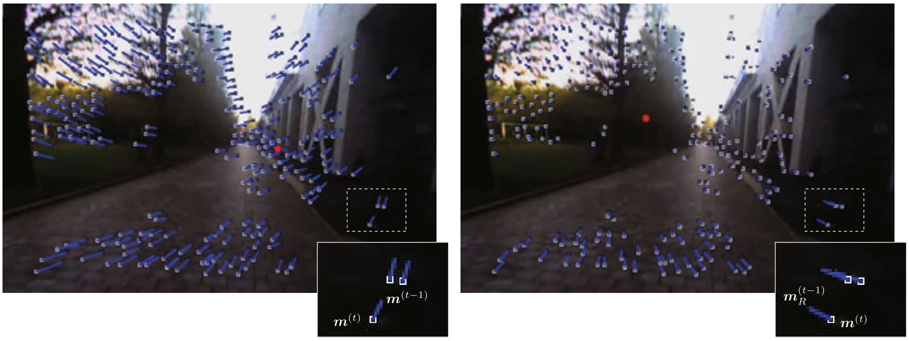
where $\boldsymbol{x}^{(t-1)}$ and $\boldsymbol{x}_R^{(t-1)}$ indicate the normalized image coordinates of the feature point. As discussed above, the following relationship also holds:

$$\boldsymbol{x}_R^{(t-1)} \sim \boldsymbol{R}\boldsymbol{x}^{(t-1)}, \tag{5}$$

and hence $\boldsymbol{m}_R^{(t-1)}$ can be written as

$$\boldsymbol{m}_R^{(t-1)} \sim \boldsymbol{A}\boldsymbol{R}\boldsymbol{A}^{-1}\boldsymbol{m}^{(t-1)}. \tag{6}$$

By applying Eq. (6) to all coordinates $\boldsymbol{m}^{(t-1)}$ of inlier flows, the translational components of flow $\boldsymbol{m}^{(t)} - \boldsymbol{m}_R^{(t-1)}$ can be computed. The FOE is computed as the point with the minimum Euclid distance to all the translational flows. Fig. 4 shows a example of all feature flows and the separated translational flows. The bright rectangles indicate feature positions in current frame $\boldsymbol{m}^{(t)}$, and the dark rectangles indicate feature positions in original image $\boldsymbol{m}^{(t-1)}$ (a) and rotated image $\boldsymbol{m}_R^{(t-1)}$. The circles overlaid in the images indicate the computed FOE.



(a) Raw feature flows

(b) Translational flows

Fig. 4. Separation of translational flows. By rotating previous frame using rotation matrix $\boldsymbol{R}$, translational flows (b) can be obtained from raw feature flows (a). Each image shows current frame $\mathrm{I}^{(t)}$. Bright rectangles indicate feature positions in current frame $\boldsymbol{m}^{(t)}$, and dark rectangles indicate feature positions in original image $\boldsymbol{m}^{(t-1)}$ (a) and rotated image $\boldsymbol{m}_R^{(t-1)}$. Circles overlaid in images indicate FOEs that are computed as point with minimum Euclid distance to all translational flows.

The above process computes the FOE based only on two successive frames; however, using multiple video frames will lead to more accurate computation of the moving direction. For this reason, we computed the FOEs between all K pairs of $\mathrm{I}^{(t)}$ and $\mathrm{I}^{(t-k)}$ ($k = 1, 2, \ldots, K$, and $K = 15$ in this work). A motion-based attention map is generated from the K FOEs by Gaussian kernel density estimation.

2.3 Aggregation of Maps

The bottom-up visual saliency maps and the egomotion-based attention maps are then aggregated to compute the final attention map. All maps are summed with equal weights, and the summed map is then normalized to have fixed maximum and minimum values. Fig. 5 shows some examples of visual saliency maps, attention maps, and the final aggregated map. We evaluated three combinations of the maps in this work: A) saliency + rotation + translation, B) saliency + rotation, and C) saliency + translation. This is further discussed in Section 3.

3 Experiments

Here, we describe the details on the experiments we carried out to evaluate what effect using motion-based attention maps had. We used a head-mounted gaze tracker to capture real egocentric videos and ground-truth gaze points. The prediction accuracy of the attention maps was assessed with the receiver operating characteristic (ROC) curves of the maps similarly to evaluating visual saliency maps. The prediction accuracy of our maps was also compared with a simple centering bias map to further demonstrate the efficiency of our method.

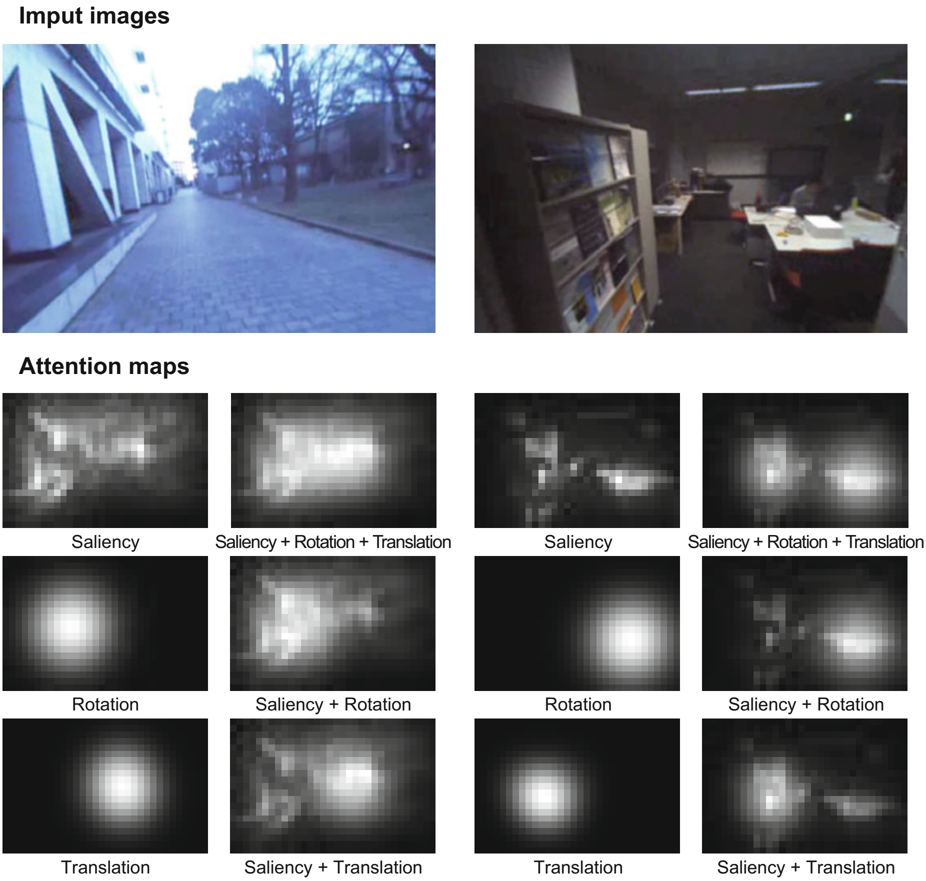


Fig. 5. Examples of attention maps. Top row shows input images, and other images show examples of visual saliency maps (saliency), motion-based attention maps (rotation and translation), and three different types of their combinations.

3.1 Experimental Settings

A mobile gaze tracker, the EMR-9 [19] developed by NAC Image Technology, was used in the experiments. A scene camera was installed on the EMR-9 as seen in Fig. 6(a), and it captured egocentric video of the subject at 30 [Hz]. The horizontal field of view of the scene camera was 121°, and the resolution of the egocentric video was 640 × 480 [pixels]. EMR-9 also had two eye cameras and two infrared light sources, and recorded the ground-truth gaze points on the egocentric video at 240 [Hz].

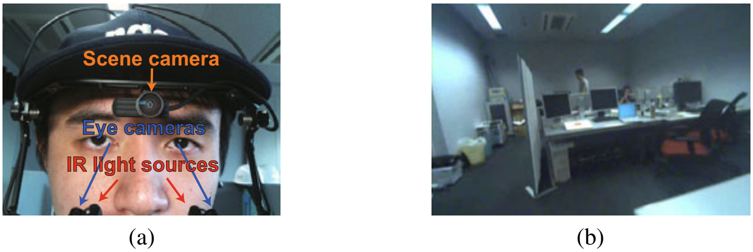


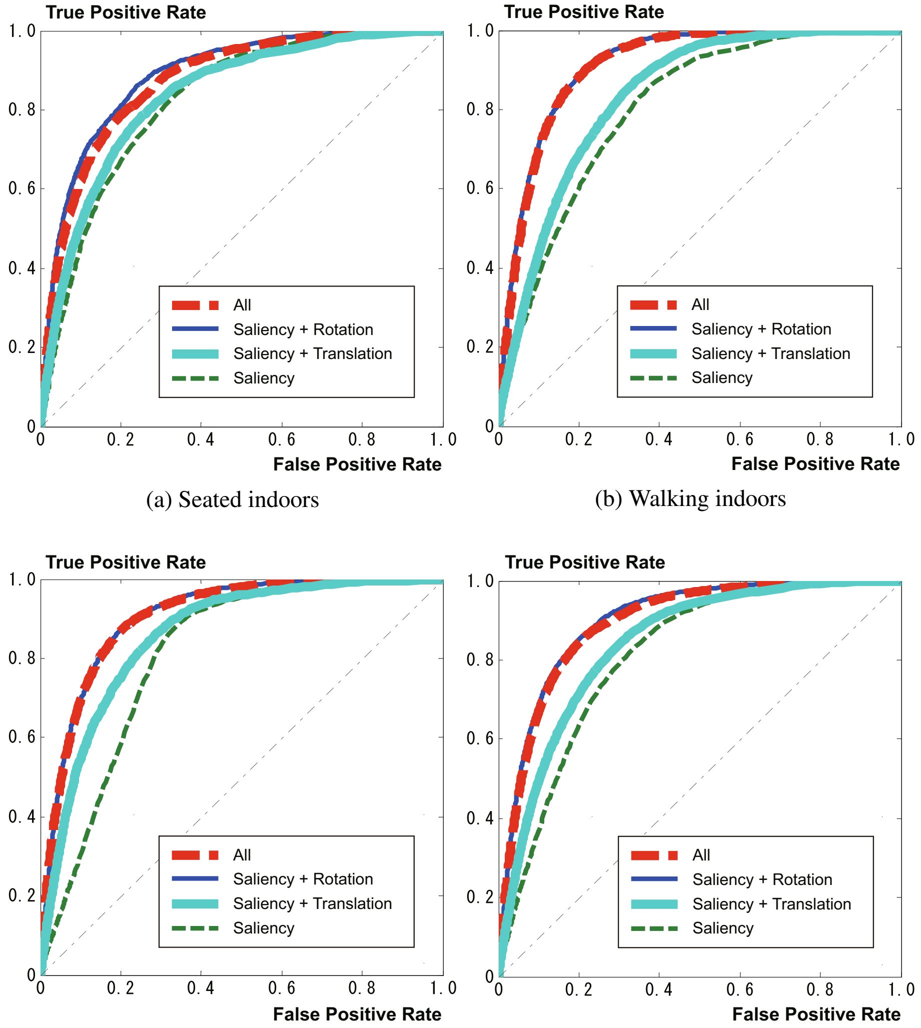
Fig. 6. (a) Mobile gaze tracker employed in experiments. Scene camera is installed and it captures egocentric video of subject at 30 [Hz]. Two eye-cameras and two infrared light sources can record ground-truth gaze points on egocentric video at 240 [Hz]. (b) Example frame of egocentric video. Horizontal field of view of scene camera was 121°, and resolution of egocentric video was 640 × 480 [pixels].

Egocentric videos and gaze points of five test subjects were recorded under three different settings in which the subjects were: seated indoors, walking indoors, and walking outdoors. Free head movements were allowed in all the settings. Fig. 6(b) shows some examples of the recorded scenes. After rejecting frames with unreliable gaze data caused by actions such as blinking and fast eye movements, the same number of $8,000$ gaze points was selected in each of the 5×3 datasets we used for evaluation.

3.2 Results

To assess how accurately the attention maps predicted a persons' visual attention, we analyzed the correspondence between the maps and the ground-truth gaze points. Fig. 7 shows the ROC curves of the attention maps generated by our framework that were drawn by sweeping the threshold value across all maps. The vertical axis indicates true positive rates, i.e., the rates of gaze points that have higher values than the threshold in the corresponding maps. The horizontal axis indicates false positive rates, i.e., rates of map regions without gaze points that have higher values than the threshold. Therefore, this indicates that the maps can predict gaze points more accurately if the curve approaches the top-left corner.

The area under the curve (AUC) values of the ROC curves are listed in Table 1, where results using a simple centering bias map (centering) have also been listed in addition to the maps (saliency, rotation, and translation) discussed above. It can be seen from these results that our proposed framework can predict actual gaze points more accurately than the standard visual saliency maps and the centering bias maps in egocentric videos. The combination of the visual saliency map and the rotation-based attention map achieved the highest AUC, and thus the highest accuracy.



(a) Seated indoors

(b) Walking indoors

(c) Walking outdoors

(d) All

Fig. 7. ROC curves of attention maps. Curves were drawn by sweeping threshold value across all maps in four datasets ((a) seated indoors, (b) walking indoorsC(c) walking outdoors, and (d) all combined). Vertical axis indicates true positive rates, i.e., rates of gaze points that have higher value than threshold in corresponding maps. Horizontal axis indicates false positive rates, i.e., rates of map regions without gaze points that have higher value than threshold.

Table 1. Prediction accuracy of attention maps. Each row lists area under curve (AUC) values of ROC curves using bottom-up visual saliency maps (saliency), rotation-based attention maps (rotation), translation-based attention maps (translation), centering bias maps (centering) and their combinations.

Method	AUC
Proposed (saliency + rotation)	**0.900**
Proposed (saliency + translation)	0.841
Proposed (saliency + rotation + translation)	0.893
Saliency	0.809
Rotation	0.892
Centering	0.884
Saliency + centering	0.890

4 Conclusion

We proposed a framework for computing human visual attention maps based on bottom-up visual saliency and egomotion. Rotation-based and translation-based attention maps were generated only using egocentric videos without requiring additional sensors. The effect of using egomotion-based maps was quantitatively evaluated using real egocentric videos, and we demonstrated that the combination of visual saliency maps and rotation-based attention maps could achieve the most accurate predictions of human attention.

Attention prediction using our framework can be done just by using egocentric videos. This has widespread possibilities for applications including casual gaze trackers and attention-based life-log systems. More sophisticated mechanisms for human egocentric visual perception will be investigated in future work to achieve more accurate prediction of visual attention.

References

1. Avraham, T., Lindenbaum, M.: Esaliency (extended saliency): Meaningful attention using stochastic image modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 32(4), 693–708 (2010)
2. Cerf, M., Harel, J., Einhäuser, W., Koch, C.: Predicting human gaze using low-level saliency combined with face detection. In: Advances in Neural Information Processing Systems (NIPS), vol. 20, pp. 241–248 (2007)
3. Costa, L.: Visual saliency and atention as random walks on complex networks. ArXiv Physics e-prints, arXiv:physics/0603025, pp. 1–6 (2006)
4. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
5. Foulsham, T., Underwood, G.: What can saliency models predict about eye movements? spatial and sequential aspects of fixations during encoding and recognition. Journal of Vision 8(2:6), 1–17 (2008)

6. Fukuchi, M., Tsuchiya, N., Koch, C.: The focus of expansion in optical flow fields acts as a strong cue for visual attention. Journal of Vision 9(8), 137a (2009)
7. Hansen, D., Ji, Q.: In the eye of the beholder: A survey of models for eyes and gaze. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 32(3), 478–500 (2010)
8. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: Advances in Neural Information Processing Systems (NIPS), vol. 19, pp. 545–552 (2006)
9. Hartley, R.: In defense of the eight-point algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 19(6), 580–593 (1997)
10. Hillaire, S., Lécuyer, A., Breton, G., Corte, T.R.: Gaze behavior and visual attention model when turning in virtual environments. In: Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology, VRST 2009, pp. 43–50. ACM, New York (2009)
11. Hillaire, S., Lécuyer, A., Regia-Corte, T., Cozot, R., Royan, J., Breton, G.: A real-time visual attention model for predicting gaze point during first-person exploration of virtual environments. In: Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology, VRST 2010, pp. 191–198. ACM, New York (2010)
12. Itti, L.: Quantitative modeling of perceptual salience at human eye position. Visual Cognition 14(4), 959–984 (2006)
13. Itti, L., Baldi, P.F.: Bayesian surprise attracts human attention. In: Advances in Neural Information Processing Systems, NIPS 2005, vol. 19, pp. 547–554 (2006)
14. Itti, L., Dhavale, N., Pighin, F., et al.: Realistic avatar eye and head animation using a neurobiological model of visual attention. In: SPIE 48th Annual International Symposiumon Optical Science and Technology, vol. 5200, pp. 64–78 (2003)
15. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 20(11), 1254–1259 (1998)
16. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: IEEE International Conference on Computer Vision (ICCV), pp. 2106–2113. IEEE (2009)
17. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. Human Neurobiology 4(4), 219–227 (1985)
18. Ma, Y., Hua, X., Lu, L., Zhang, H.: A generic framework of user attention model and its application in video summarization. IEEE Transactions on Multimedia 7(5), 907–919 (2005)
19. nac Image Technology Inc.: Emr-9, http://www.nacinc.com/products/Eye-Tracking-Products/EMR-9/
20. Parkhurst, D., Law, K., Niebur, E.: Modeling the role of salience in the allocation of overt visual attention. Vision Research 42(1), 107–123 (2002)
21. Qiu, X., Jiang, S., Liu, H., Huang, Q., Cao, L.: Spatial-temporal attention analysis for home video. In: IEEE International Conference on Multimedia and Expo (ICME 2008), pp. 1517–1520 (2008)
22. Shi, J., Tomasi, C.: Good features to track. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 593–600. IEEE (1994)
23. Tomasi, C., Kanade, T.: Detection and tracking of point features. Carnegie Mellon University Technical Report CMU-CS-91-132, pp. 1–22 (1991)
24. Treisman, A., Gelade, G.: A feature-integration theory of attention. Cognitive Psychology 12(1), 97–136 (1980)
25. Wang, W., Wang, Y., Huang, Q., Gao, W.: Measuring visual saliency by site entropy rate. In: Computer Vision and Pattern Recognition (CVPR), pp. 2368–2375. IEEE (2010)
26. Yamada, K., Sugano, Y., Okabe, T., Sato, Y., Sugimoto, A., Hiraki, K.: Can saliency map models predict human egocentric visual attention? In: Proc. International Workshop on Gaze Sensing and Interactions (2010)

FAW for Multi-exposure Fusion Features

Michael May, Martin Turner, and Tim Morris

The University of Manchester, UK
michael.may@student.manchester.ac.uk,
{martin.turner,tim.morris}@manchester.ac.uk
http://www.cs.manchester.ac.uk
http://www.michael-may.co.uk

Abstract. This paper introduces a process where fusion features assist matching scale invariant feature transform (SIFT) image features from high contrast scenes. FAW defines the order for extracting features: features, alignment then weighting. The process uses three quality measures to select features from a series of differently exposed images and select a subset of the features in favour of those areas that are defined as well exposed from the different images. The results show an advantage in using these features over features extracted from the common alternative techniques of exposure fusion and tone mapping which extract the features as AWF; alignment, weighting then features. This paper also shows that the process allows for a more robust response when using misaligned or stereoscopic image sets.

Keywords: feature fusion, SIFT, HDR, LDR, tone mapping, exposure fusion, stereo.

1 Introduction

Feature matching is a common computer vision application. In high contrast lighting conditions it can be difficult to extract features in all areas of a scene with a single exposure image as areas can be over or under exposed. As such, vital information about a scene can be missed. The problem that this paper solves is how to best utilise multiple exposure images to match features in scenes with a large dynamic range. The main contribution of this paper is a feature fusion process using the scale invariant feature transform (SIFT) within sets of images taken of the same scene with varied exposures. These features cover a larger dynamic range in a scene and are extracted in a way which improves match accuracy when compared to extracting features directly from high dynamic range image types. **FAW** defines the recommended order for extracting fusion features; **F**eatures extraction, image **A**lignment then pixel **W**eighting. This is opposed to **AWF**, the order for generating tone mapped and exposure fusion images and extracting featrues from them; **A**lignment of the images, pixel **W**eighting and image merging and then **F**eatures extraction.

The concept is based on exposure fusion [14,15] and its purpose is to create an improved set of features which represent a higher dynamic range then a

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 289–300, 2011.

set of features extracted from a single image. A key component is that areas which contain information unseen in one exposure can utilise the features from a differently exposed image. The process selects from the best exposed areas of each exposure image using three different measures given in Sect. 2. This generates a new set of features which cover a larger dynamic range. This process can be applied to aligned images, as with exposure fusion, but can also be extended to misaligned and stereoscopic images as shown in Sect. 3.

1.1 Scale Invariant Feature Transform

The SIFT feature detection algorithm, developed by David Lowe [9,10], is a four stage process that extracts highly descriptive features from an image. The features are invariant to rotation and robust to changes in scale, illumination, noise and small changes in viewpoints. The features can be used to indicate if there is any correspondence between areas. The four stages of the SIFT algorithm are as follows:

1. Scale-space extrema detection. **2.** Feature localisation and selection.
3.Orientation assignment of features. **4.** Creation of the descriptor vector.

To match features the Euclidean distance between two feature vectors is used to find the nearest neighbour. The ratio between the best and second best match is used to confirm a match.

1.2 High Dynamic Range Images

Dynamic range is the ratio between the brightest and darkest pixels in a scene. High dynamic range (HDR) images often consist of three 32-bit floating point numbers [17], one per channel, whereas low dynamic range (LDR) images use 8-bits per channel. Data outside the range is truncated to the nearest value so information may be lost. For LDR photography an exposure must be selected to attempt to capture the most important information within the limited dynamic range of the camera which is not always possible. In terms of SIFT features, it has been shown [12] that extracting the information from the dark and bright areas as well means that there is a higher likelihood of locating the object of interest due to the higher number of stable features available.

HDR images are generally generated from multiple bracketed LDR images of the same scene taken in quick succession at different exposures [1,11]. The response function of the camera is computed, which maps the pixel value stored in an image to the radiance in a scene. Using this and a weighting function, which reduces the contribution of points at the edges of the dynamic range of the LDR image, a HDR image can be created. The HDR image contains the best exposed areas displaying high detail from the most appropriate LDR images.

1.3 Tone Mapping

It is impossible to display HDR images on most displays as the dynamic range of the average monitor is only 2 orders of magnitude [17]. Tone-mapping has been developed to convert a HDR image into an 8-bit LDR format so that they can be viewed on a conventional display.

Techniques have been proposed for both global and local tone mapping. Global operators apply a uniform remapping of the pixel intensity values to compress the dynamic range [2,3,7]. They can be faster than local operators but can fail to produce a visually pleasing image due to their inability to take account the varying responses to the algorithm on different parts of an image.

Local tone mapping algorithms [4,5,8,18,16,21,22] work by reducing the gradient magnitude in the areas of high gradient while preserving the areas of low gradient. The human visual system is insensitive to absolute brightness but responds to local contrast, meaning that global differences in brightness can be reduced so long as the darker parts of the image remain darker and the brighter parts remain brighter. These methods can preserve more detail but sometimes result in unrealistic final images.

1.4 Exposure Fusion

Exposure fusion [14,15,19] is a technique for fusing a bracketed exposure sequence into a high-quality, tone-map like image, without converting to HDR first. Its advantages over tone mapping include the fact that no HDR image needs to be computed often making the process faster and simpler. Also the process is more robust as the exposure values are not needed and a flash can be used with the camera.

The process uses weighted averages of the images where the weightings are calculated based on certain properties of the image; *Contrast*, *saturation* and *well-exposedness* (see Sect. 2). These are each weighted, combined and normalised and then used to calculate a weighted average of the exposure images' pixels to create a fusion image.

Multi-resolution fusion [14,15] is a continuation of this technique to reduce the appearance of seams in the final image. Each of the input images is decomposed into a Laplacian pyramid and the corresponding weight map is decomposed into a Gaussian pyramid. The Laplacian pyramid of the fusion image is determined by the weighted average of the input Laplacian pyramid, where the weights are given by the corresponding scale in the Gaussian weight map. Finally the fused output image can be reconstructed from its Laplacian pyramid by using an inverse transform.

2 Fusion Feature Selection

The process of selecting fusion features utilises the main measures of exposure fusion [14,15]. A set of images of varying exposures are taken and for each of these images a set of features are extracted using SIFT as shown in Fig. 1. These features are then used to accurately align the images using RANSAC [20]. The feature locations are also transformed to match the transformation of the images. For each pixel in the aligned images weightings are generated

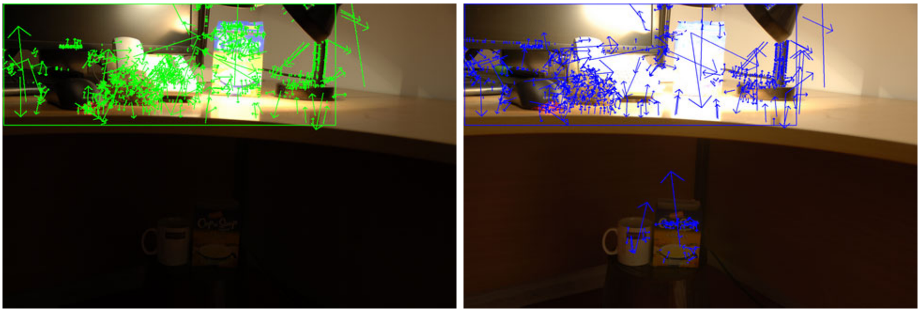


Fig. 1. An example of two aligned input images taken at different exposures. The arrows represent the scale, orientation and position of the SIFT features. The bounding box in each shows the areas within which SIFT features have been matched between the images using RANSAC during the alignment process [20].

using some or all of the three measures outlined below. The weightings for each pixel indicate the exposure image in which each pixel is best exposed. This is then used to select which features are added to the set of fusion features using a Gaussian weighting at the scale and radius of the feature. **FAW** defines this order; **F**eatures extraction, image **A**lignment and then pixel **W**eighting. This is opposed to **AWF**; **A**lignment of the images, pixel **W**eighting and merging the images and then **F**eatures extraction. This is used for matching tone mapping and exposure fusion images. This process has been briefly outlined previously by May et al. [12] using only the contrast measure (C).

Contrast Measure C: The gradient magnitude $m\left(x,y\right)$ is calculated across the image, F, for each greyscale pixel location:

$$m\left(x,y\right)=\sqrt{\left(F\left(x+1,y\right)-F\left(x-1,y\right)\right)^{2}+\left(F\left(x,y+1\right)-F\left(x,y-1\right)\right)^{2}} \quad (1)$$

This gives larger values for textured areas and this indicates if an area of the image is well exposed as over or under exposed areas will have small gradient values. Using the absolute values returned by a Laplacian filter as suggested by the Mertens et al. [14,15] has been replaced by the gradient magnitude. Using a zero crossing, second derivative, function to calculate the weighting means that the edge peaks will return a value of zero. Thus, two edges, one with a large magnitude and one which is much smaller in magnitude will both have a value of 0 at their apex and a weighting based on this will weight both pixel values equally. If they are slightly misaligned then one edge pixel will get the full weighting in its favour at a point when the other image may have larger edge. A first derivative function returning the gradient magnitude allows edge gradients values to be compared and weighted accordingly.

Satuaration Measure S: As an image is exposed for a longer period of time it becomes desaturated. The less saturated the image, the more washed-out it appears until finally, when saturation is at zero, the image becomes a monochrome or greyscale image. This is used as another measure of how well exposed the image is. The standard deviation of the three RGB values is calculated at each pixel to generate this measure.

Well-Exposedness Measure E: This is a measure to weight the value based on its closeness to the maximum or minimum pixel values. Well exposed parts of an image will consist of pixel values close to 0.5 and as values get closer to zero or one they indicate under and over exposed areas. A Gaussian function is used to calculate a weighting w for each colour channel intensity i independently at each pixel and the values are multiplied to generate the final weighting E. A σ value of 0.2 is used as suggested by Mertens et al. [14].

$$w = \exp\left(-\frac{(i-0.5)^2}{2\sigma^2}\right) \tag{2}$$

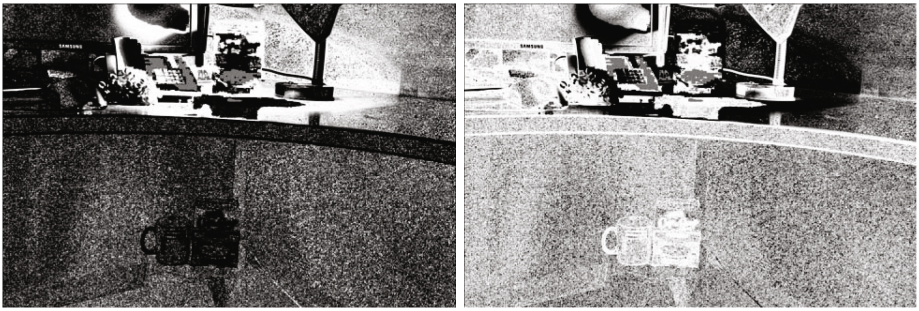


Fig. 2. The normalised weightings generated from the exposure measures for the images in Fig 1. Darker values indicate a higher weighting and indicate the areas from each image which are better exposed.

A subset of all of the image measures can be used to select a preferred set of features. If more than one measure is used they are combined by multiplying and each can be weighted to vary the effect of each measure. For this paper all three measures are used and weighted equally. Each aligned exposure image will then have its own set of pixel value weightings. The weighting are normalised to the range of 0 to 1 for the corresponding pixels in each exposure image as shown in Fig. 2.

To select the features for the final set the weightings at each feature location are used. Only the features from the best exposed locations will be preserved. The selection takes place over the area and scale that the feature was originally extracted. At each location at the scale of the feature, σ is used to calculate an approximate radius of the feature; 6σ [10]. A Gaussian weighting of that radius and with a standard deviation corresponding to the scale of the feature σ is then applied to the weights centred on the feature position. The resultant values are summed across the total feature area and used to select the feature. A feature is selected if the summed value is greater than that for the same location in all the other images. The final set of features is shown in Fig. 3.

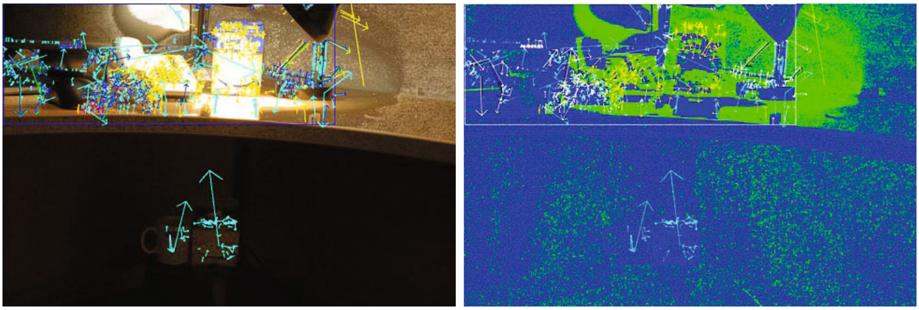


Fig. 3. The set of fusion features displayed on a rough exposure fusion image on the left and on a binary fusion image on the right. The binary image shows which areas are best exposed in each image and relate to the feature colours used in Fig. 1. The yellow arrows indicate features selected from image 1 and the turquoise features are selected from image 2. The blue and green arrows are from the features which match between the images and have been blended for the final feature set.

2.1 Feature Blending

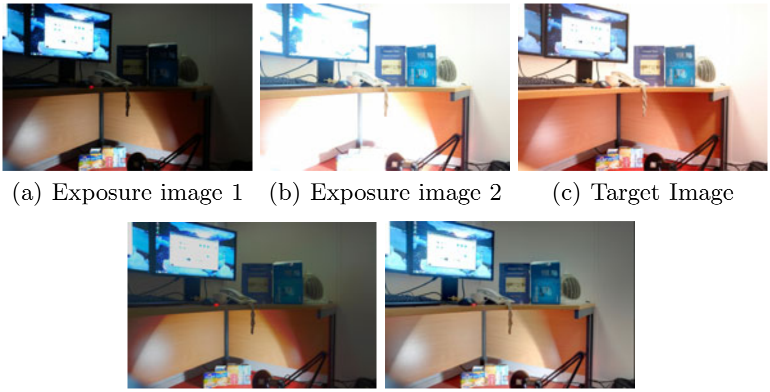
The image alignment process uses RANSAC [20] to register matched features and calculate a transform to align the images. The features which are successfully aligned between images can be merged for the final fusion feature set by averaging their vectors as they both must be in well exposed areas for them to match. The alternative is to treat these features like any other and select one based on their weightings.

2.2 Evaluation

The scenario for testing the feature fusion process is as follows:

A high contrast scene is obtained by using a spotlight in a darkened room or locating an area of shadow. Two aligned exposures of the scene are captured, each exposed correctly for the different parts of the scene. A third, target image, is captured. This is done by taking a picture of the scene after the scene lighting has been changed by turning on a larger brighter light source (the camera flash or ceiling light) which allows the whole scene to be captured in a single LDR exposure. Neither exposure image will match to all of the areas of the target image but a high dynamic range image created from both images should. This scenario relates to a real world scenario in which a well-lit target image has been captured under controlled circumstances and an attempt is being made to locate an object or scene where the dynamic range is large.

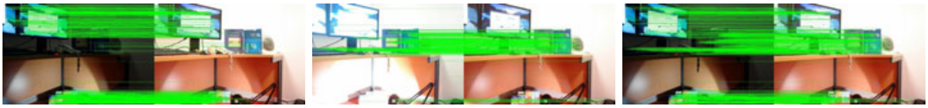
The two exposure images are used to create a tone mapped image using Devlin's [4] and Reinhard's [16] techniques and an exposure fusion [14] image is also generated as shown in Fig. 4. If the exposure images are misaligned they are aligned first to get the best possible results [20]. A set of SIFT features are extracted from each resultant HDR representation. These processes represents the AWF paradigm as they are ordered; alignment, weighting and then features.



(a) Exposure image 1 (b) Exposure image 2 (c) Target Image

(d) Tone Map (e) Exposure Fusion

Fig. 4. Example set of high contrast images used for testing



(a) Exposure image 1 (330) (b) Exposure image 2 (297) (c) Fusion features (442)

Fig. 5. Feature matching examples represented by the parallel lines. The number in brackets gives the number of matches. Note that there are more fusion features matches.

The two exposure images are used to create a set of fusion features. The three sets of features are matched to the target image using the nearest and second nearest neighbour technique described by Lowe [10]. All the features from both LDR exposure images are also matched for comparison as shown in Fig. 5.

2.3 Results

Thirty one aligned exposure pairs were used and Tab. 1 shows the average results of matching to the target images. They show that fusion features perform better than the synthetic images generated from exposure fusion and tone mapping in high contrast scenarios. FAW has an advantage over AWF.

For the aligned image tests Tab. 1 shows that a higher percentage of the features match from the fusion feature set. The correspondence ratio [13] is 40%

Table 1. The mean results for 31 test exposure image pairs showing the number of features extracted, the matched features and the correspondence ratio (number of matches/total features) [13].

	Fusion Features	Tone Map	Exposure Fusion	All Features
Total Features	1165	1848	1690	2470
Matched Features	170	138	183	302
Correspondence Ratio	0.14	0.08	0.10	0.12

greater than for exposure fusion, 75% greater than for tone mapping and 16% greater than if all the features are matched. The correspondence ratio provides a good indication of whether the images match well. Using the number of matched features as an indicator is unreliable as one image may have more matches but if it has a higher number of total features then there is an increased chance of false positives.

The results show that the feature set for feature fusion is generally smaller and there are fewer superfluous features. Exposure fusion generates, on average, 45% more features but only generates 8% more feature matches therefore the extra features provide little advantage. Of the 31 test cases the exposure fusion had the highest correspondence ratio in 23 cases, the tone mapped images in 4 cases and the exposure fusion images in 3 cases. In 1 case matching was unsuccessful in matching any features for all three feature types.

3 Stereo Fusion Features

Stereoscopic systems are common in computer vision applications. To utilise this and extend the dynamic range of such systems it is proposed that the two cameras have different exposures values (EVs) resulting in a lower quality 3D reconstruction but increasing the dynamic range for feature matching. This may be preferable in some circumstances where an increased feature matching range is desirable over high quality 3D. Stereo fusion features is the process of generating fusion features from misaligned stereo images of varied exposure.

When using stereo images to create tone maps often, after warping, the images do not align correctly. This is due to the absence of a homography which will correctly warp all areas of the image and leads to ghosting and edge effects which means that features extracted from a synthetic image generated from these pairs may contain erroneous features. Fig. 6 demonstrates the problem. Since the fusion feature process doesn't generate new images or features this problem is negated.

A compromise can be made between good 3D and good HDR images by varying the exposure difference and baseline of the stereo images. A stereo pair with a small baseline will generate a poor 3D representation but will allow the

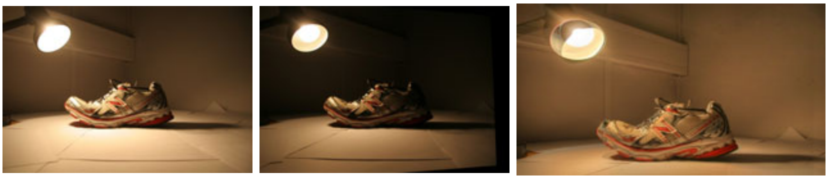


Fig. 6. A pair of stereo images at 10° and a 2 EV difference, the second is warped to align with the first. The tone map image generated on the right hand side demonstrates the ghosting and other artefacts generated by tone mapping stereo images. Selecting SIFT features directly from the tone map can therefore generate unreliable features.

images to more easily registered for HDR. A large baseline has the opposite effect. The exposure difference between the stereo cameras has an effect as a large difference will make the dynamic range of the features increase but make it more difficult to match features between the images. This is shown in Fig. 7.

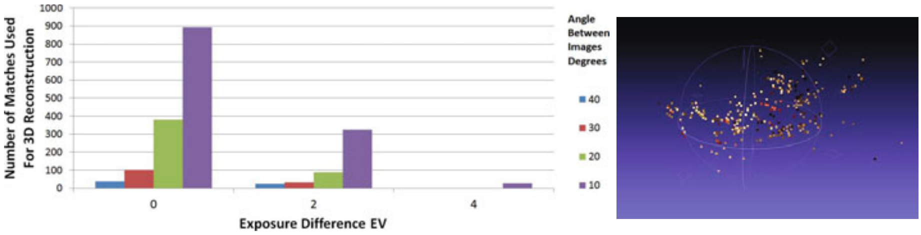


Fig. 7. A graph showing the number of feature that are used to create a set of 3D points from stereo pairs at various angles and exposures. The exposure axis values represent the change in EV between the image pair. The data has been generated from 12 pairs of images, similar to those in Fig. 8, using Bundler [6]. As the exposure and angle difference increases the number of features that can be matched to create the cloud decrease. This demonstrates the trade-off between the number of reliable 3D features and the dynamic range captured.

Bundler [6], a structure from motion tool which utilises bundle adjustment, can be used to generate a 3D model of the features and indicate which features can be aligned, Fig. 8. This subset can be used for the projective transformation from one image to the other. If the 3D data is not required RANSAC alone [20] can be used for alignment as in the initial example. The second image is transformed to align with the first. Features which can be aligned with an projective transformation are surrounded by a bounding box, Fig. 8, and features

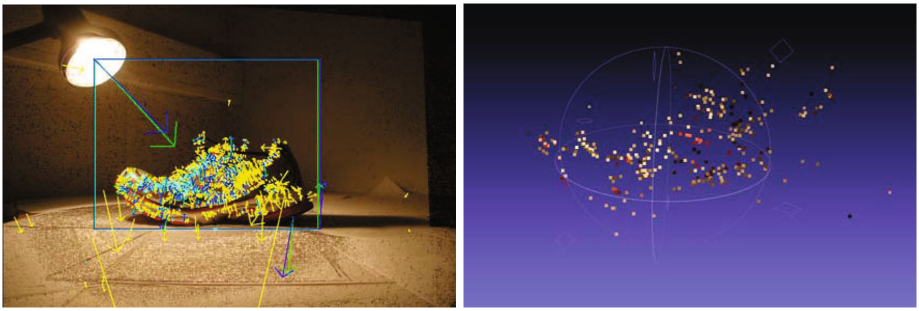


Fig. 8. The set of features selected from two stereo input images in Fig. 6. A lower quality 3D point cloud is generated than if the EV values were the same but the dynamic range of the feature set is higher. Features have been selected from the second image on the toe area of the shoe which is over exposed in the first image because of the light shining on it.

outside this area may be inaccurate. This area decreases as the EV or baseline increase. The fusion features process is then completed as before. The features produced using stereo images will provide information about the presence of that object in a scene. Features outside the bounding box are unreliable in their exact location due to the lack of an projective transformation which will accurately transform all the feature locations from the second image to the first. Localisation can then rely solely on the features which match from the first image and those within the aligned area.

3.1 Results

The evaluation has been conducted in a similar manner to the standard fusion feature tests. A set of twenty eight stereo images have been used are the full set of images shown in Fig. 6. They consist of the stereo pairs taken at measured exposures and angles. The second image and its feature positions are warped to best align to the first before exposure fusion takes place. The results are shown in Fig. 9. In all cases the greater correspondence ratio [13] for feature fusion demonstrates the advantages over the exposure fusion and tone mapped techniques.

4 Analysis

The results clearly show the advantages of using the fusion features and FAW over the synthetic images and AWF for these test cases. This is due to the artefacts, compression and changes in luminance which occurs when the synthetic images are created. Any slight misalignment can affect the resultant SIFT features whereas the fusion features are more robust to these errors. The fact that

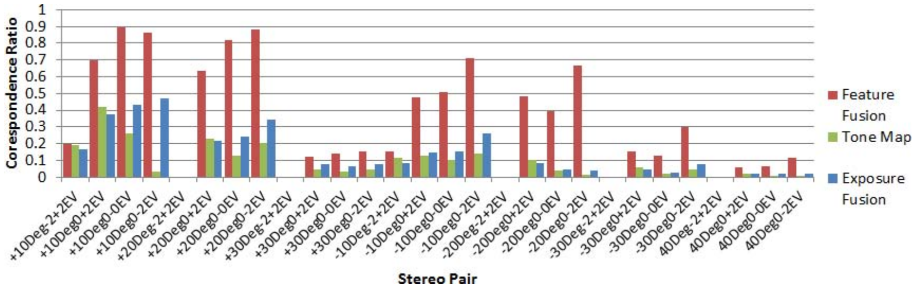


Fig. 9. A graph showing the correspondence ratio (number of matches/total features) for fusion features and features extracted from exposure fusion images generated from 28 pairs of stereo images. The x-axis shows the stereo pair disparity in degrees (plus or minus refers to left or right of the first image) and the EV of the two images. The features are all matched to a single target image taken at 0° and 0 EV at approximately 1 foot away from the shoe. The images used are the same as those used for Fig. 7 and resemble those shown in Fig. 6.

the fusion feature process relies on features which have been extracted from scene images with fewer processing stages. The weighted pixel averaging that takes place in the exposure fusion and tone mapping processes effects the quality of the pixel values as poorly exposed areas can still negatively affect the final, average, pixel values.

The difference between the feature fusion and other results for the stereo test cases is because of the substantial ghosting effects which are exaggerated as the stereo baseline is increased. The advantage of the stereo tests is more useful in the lower baseline examples where the images align well with an projective transformation and as such the use of the feature fusion technique is valid. As the angle increases the 3D object cannot be satisfactorily aligned with a projective transformation and as such aligned areas of the images which represent the same positions in space become smaller thus the fusion feature technique becomes less reliable. As such the area from which fusion features are selected can be limited to a bounding box.

5 Conclusion

The process introduced in this paper allows sets of features to be generated which allow matching to take place in high contrast environments. This is advantageous as it allows objects to be detected using features which may otherwise be hidden in a single exposure image. The performance advantage of using the fusion feature technique has been demonstrated over extracting features from exposure fusion or tone mapped images. This is due to the artefacts and changes that are introduced to these synthetic images which create features which do not always match to features taken from images captured directly from a scene. The advantages of FAW over AWF are clear as FAW reduces artefacts introduced in the image processing stages.

Other advantages of using the process include the robustness to misaligned 3D images at small changes for non-projective scenes. Misaligned images will make noisy tone maps and exposure images but using the fusion as a way of selecting features is better than trying to generate new ones. The process generates a subset of the total features and generally generates fewer features then the synthetic techniques so faster matching can take place. Fusion features doesn't require a HDR image to be generated therefore doesn't require as many, resource consuming, intermediate steps.

Future work will include comparison to other tone mapping operators and testing other combinations of fusion feature quality measures.

References

1. Debevec, P., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: ACM SIGGRAPH Classes, pp. 1–10. ACM (2008)
2. Devlin, K., Reinhard, E.: Dynamic Range Reduction Inspired by Photoreceptor Physiology. IEEE TVCG 11(1), 13–24 (2005)

3. Drago, F., Myszkowski, K., Annen, T., Chiba, N.: Adaptive logarithmic mapping for displaying high contrast scenes. CGF 22, 419–426 (2003)
4. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. ACM TOG 21, 257–266 (2002)
5. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. ACM TOG 21(3), 249–256 (2002)
6. Helmer, S., Meger, D., Muja, M., Little, J.J., Lowe, D.G.: Multiple Viewpoint Recognition and Localization. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 464–477. Springer, Heidelberg (2011)
7. Larson, G., Rushmeier, H., Piatko, C.: A visibility matching tone reproduction operator for high dynamic range scenes. IEEE TVCG 3(4), 291–306 (1997)
8. Li, Y., Sharan, L., Adelson, E.: Compressing and companding high dynamic range images with subband architectures. ACM TOG 24, 836–844 (2005)
9. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV, vol. 2, p. 1150 (1999)
10. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
11. Mann, S., Picard, R., Section, Massachusetts Institute Technology Perceptual Computing: On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures (1995)
12. May, M., Morris, T., Markham, K., Crowther, W.J., Turner, M.J.: Towards Object Recognition using HDR Video, Stereoscopic Depth Information and SIFT. In: EG UK TPCG (2009)
13. May, M., Turner, M.J., Morris, T.: Analysing False Positives and 3D Structure to Create Intelligent Thresholding and Weighting Functions for SIFT Features. In: Ho, Y.-S. (ed.) PSIVT 2011, Part I. LNCS, vol. 7087, pp. 191–202. Springer, Heidelberg (2011)
14. Mertens, T., Kautz, J., Van Reeth, F.: Exposure fusion: A simple and practical alternative to high dynamic range photography. CGF 28, 161–171 (2009)
15. Mertens, T., Kautz, J., Van Reeth, F.: Exposure fusion. In: PG. pp. 382–390. IEEE (October 2007)
16. Reinhard, E.: Dynamic range reduction inspired by photoreceptor physiology. IEEE TVCG 11(1), 13–24 (2005)
17. Reinhard, E.: High dynamic range imaging: acquisition, display, and image-based lighting. Morgan Kaufmann (2006)
18. Reinhard, E., Stark, M., Shirley, P., Ferwerda, J.: Photographic tone reproduction for digital images. ACM TOG 21(3), 267–276 (2002)
19. Tico, M., Gelfand, N., Pulli, K.: Motion-blur-free exposure fusion. In: IEEE ICIP, pp. 3321–3324, No. I (2010)
20. Tomaszewska, A., Mantiuk, R.: Image registration for multi-exposure high dynamic range image acquisition. In: WSCG, pp. 49–56 (2007)
21. Tumblin, J., Rushmeier, H.: Tone reproduction for realistic images. IEEE CGA 13(6), 42–48 (1993)
22. Xiao, F., DiCarlo, J., Catrysse, P., Wandell, B.: High dynamic range imaging of natural scenes. In: CIC, pp. 337–442 (2002)

Efficient Stereo Image Rectification Method Using Horizontal Baseline

Yun-Suk Kang and Yo-Sung Ho

School of Information and Communicatitions
Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Republic of Korea
{yunsuk,hoyo}@gist.ac.kr

Abstract. In this paper, we propose an efficient stereo image rectification method using the horizontal baseline. Since the stereo camera is generally manually arranged, there are geometric errors due to the camera misalignment and the differences between the camera internal characteristics. Although the conventional calibration-based stereo image rectification method is simple, it has an opportunity to provide the results that have some visual distortion such as image skewness. Therefore, the proposed method calculates the baseline for stereo image rectification, which is parallel to the horizontal line in the real world. Using this baseline, we estimate the camera parameters and the rectification transform. By applying the transform to the original images, we obtain the rectified stereo images. Experimental results show that the results of the proposed method provide the better rectified stereo image without visual distortion.

Keywords: Image rectification, stereo image, stereo camera, 3DTV.

1 Introduction

Three-dimensional (3D) TV provides us more realistic video contents than the current two-dimensional (2D) television broadcasting. Since the input signal of 3DTV is composed of more than single viewpoint images or videos, users can watch the scene with immersive feeling. In recent years, much research on 3DTV and 3D content generation has been investigated to satisfy the increasing demands for realistic multimedia services in the world [1].

In order to generate 3D contents for 3DTV, at least two view images are required basically. Two cameras, called the stereo camera, capture a 3D scene or object in the real world from two different positions. Users watch this stereo image with 3D sense with stereoscopic displays. Moreover, from this stereo image, we can estimate the scene's depth information using stereo matching [2], and also generate novel view images based on the depth.

However, there is a constraint to use stereo images for 3D applications. Two image planes of the stereo camera determine their epipolar geometry that satisfies the epipolar constraint between two images. Epipolar constraint is that a point in one image has its corresponding points in the other image along an epipolar line.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 301–310, 2011.

Therefore, if the epipolar lines in each image plane are parallel, the corresponding points have the same vertical coordinates. In other words, there is no vertical pixel difference between two images. In this case, the stereo image has only horizontal displacement. The visual quality of the image as the 3D contents increases and also the stereo matching process becomes very simple [3]. Unfortunately, the practical stereo image captured by a manually arranged stereo camera does not have the parallel epipolar lines. There are not only position and orientation differences but also internal parameter differences between two cameras.

In order to solve these problems in stereo images, we perform image rectification. Image rectification is rotation and movement of two image planes that makes epipolar lines parallel each other. The rectified stereo image is then considered as the images captured by two physically-equal cameras with only horizontal camera interval.

Image rectification has been studied for long time. There are two categories; one is based on the image features [4] [5], and the other is calibrated case [6] [7]. Recently, rectification has been extended to cover the multiple views [8]. In general, the result of image feature based rectification has some visual distortion such as image skewing. It is also influenced by the extracting features. While the calibration based image rectification gives more stable results and rectified camera parameters which is essential information for 3D applications. However, the stereo cameras have to be calibrated before the rectification, and the reliability of calibration is also influence on the results.

In this paper, we explain a stereo image rectification method using the horizontal baseline. We introduce the stereo geometry briefly in Section 2. In section 3, we explain the proposed method. After scene capturing and camera calibration, we calculate the horizontal baseline for rectification. Using this baseline, we estimate the rectification transform. By applying this transform to the original stereo image, we obtain the rectified stereo image. After showing the experimental results in Section 4, we conclude this paper in Section 5.

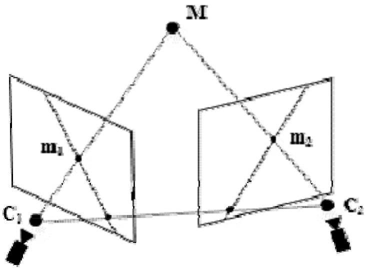


Fig. 1. Stereo geometry

2 Stereo Geometry

Figure 1 shows the geometry of stereo image. There are two camera centers $\mathbf{C}_1$ and $\mathbf{C}_2$, and the object at the point $\mathbf{M}$ in the 3D space. These three points are in the world coordinate system, and two cameras also have their own coordinate systems. Each camera coordinate system has three axes; vertical, horizontal, and principal axes. The principal axis is also called the optical axis, which indicates the optical ray direction of the camera.

By operating two cameras, the object located on **M** is projected to the image points $\mathbf{m}_1$ and $\mathbf{m}_2$ in each image plane. The corresponding point of $\mathbf{m}_1$ in the right image plane has to be on the epipolar line. The epipolar line of the right image plane is defined as the intersecting line between the right image plane and the plane described by **M**, $\mathbf{C}_1$, and $\mathbf{C}_2$, which is called the epipolar plane.

Figure 2 shows the geometry of the rectified stereo image that has the parallel epipolar lines in each image plane. All the points in the Fig. 2 excluding **M** are changed and also each image plane is rotated and moved. Therefore, two image planes and epipolar lines are parallel to the line through $\mathbf{C}'_1$ and $\mathbf{C}'_2$ which is called the baseline. In this case, two image points $\mathbf{m}'_1$ and $\mathbf{m}'_2$ have the same vertical coordinates. It means that there is no vertical pixel displacement between two corresponding points.

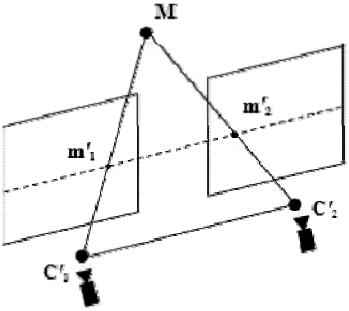


Fig. 2. Rectified stereo geometry

3 Proposed Stereo Image Rectification Method

In this section, we explain our proposed method to rectify the stereo image. In the conventional method based on the camera parameters [6], the baseline for rectification is determined as the line through $\mathbf{C}_1$ and $\mathbf{C}_2$ in Fig. 1. Then, $\mathbf{C}'_1$ and $\mathbf{C}'_2$ are the same as $\mathbf{C}_1$ and $\mathbf{C}_2$ after rectification. If this baseline is not parallel to the horizontal line in the real world, the rectified image can be skewed with respect to the users' view.

Therefore, the proposed method calculates the baseline that is parallel to the horizontal line in the real world. Figure 3 shows the procedure of the proposed method. After scene capturing, we estimate the camera parameters by camera calibration [9]. By using these camera parameters, we calculate the baseline which is parallel to the real horizontal line, and then we estimate the camera parameters of the rectified stereo image based on the baseline. Finally, we obtain the rectified stereo image by applying the rectification transform to the captured images. This transform is computed using both of the original and estimated camera parameters.

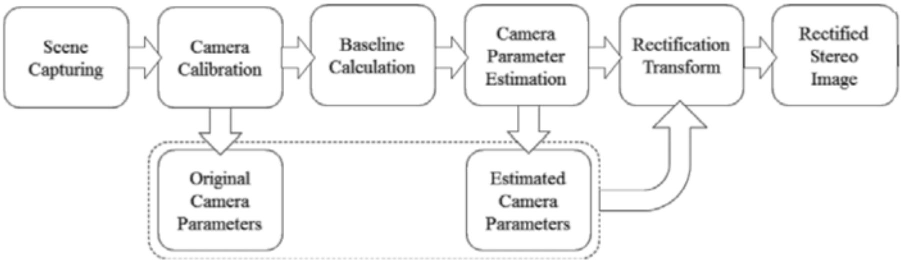


Fig. 3. Procedure of the proposed method

3.1 Baseline Calculation

After obtaining camera the parameters by camera calibration, we calculate the baseline. Baseline calculation begins with the initial line which is obtained by connecting the two camera centers. From this initial line, we can calculate the baseline. The baseline must satisfy the following two conditions. First, this baseline and the initial line are on the same plane that has its normal vector as the direction of the new principal axis. The new principal axis is determined as the direction orthogonal to both of the initial line and the average direction of all the original vertical axes. It means that the baseline can preserve the orientation of the camera array which is obtained based on camera positions.

The second condition is that the baseline is parallel to the horizontal line in the real world. It guarantees that the rectified stereo image according to this baseline does not have the skew problem. In order to obtain such a baseline, we use a line image projection algorithm that requires an image containing a short and non-tilted line like Fig. 4(a). Through the line image projection, we can measure the slope of the initial line, and then we can calculate a suitable correction vector to make the baseline parallel to the real horizontal line.

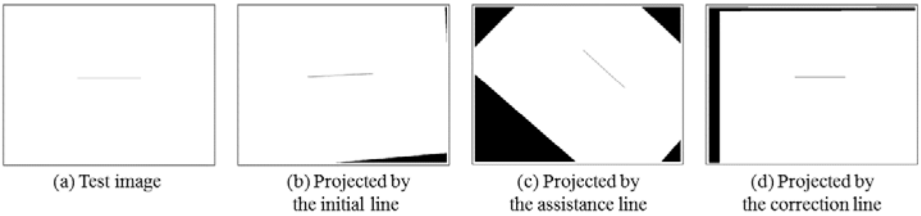

(a) Test image
(b) Projected by the initial line
(c) Projected by the assistance line
(d) Projected by the correction line

Fig. 4. Line image projection

In order to measure the slope of the initial line, we assume that the line image is left of the stereo view. We then project this image so that the horizontal axis of the image plane is parallel to the initial line. As a result, the projected image has the line tilted as the slope of the initial line as indicated in Fig. 4(b). We can measure the slope of this line by counting the number of pixels between the start-point and the end-point of the line. This measured value means the slope of the initial line.

After measuring the slope of the initial line, we need a correction vector to obtain the baseline that satisfies the second condition. Figure 5 shows how to calculate the correction vector. The cross product between the new principal axis and the initial line vector makes the orthogonal vector. The correction vector is then calculated as the sum of the initial line vector and the orthogonal vector.

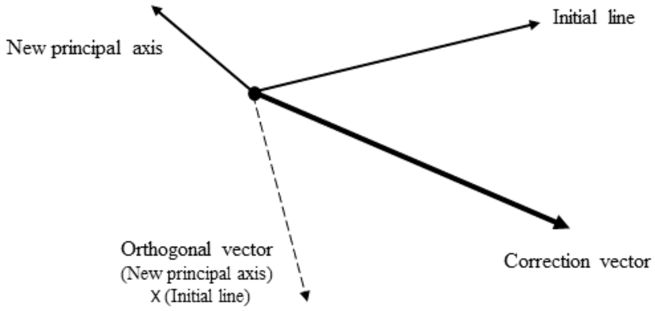


Fig. 5. Correction vector calculation

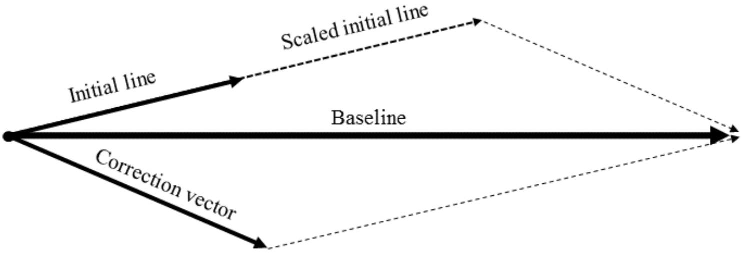


Fig. 6. Baseline calculation

In order to calculate the baseline, we measure the slope of the correction vector. By using the line image projection again, the projected image in accordance with the correction vector has an opposite slope to the initial line like Fig. 4(c). Finally, we can calculate the baseline which is parallel to the real horizontal line by summing the initial line vector and the correction vector with a proper scale factor. Let *i* and *c* be the slopes of the initial line vector and the correction vector, respectively. The scale factor s is defined as the ratio of i and c. The baseline vector $\vec{\mathbf{b}}$ is then calculated as Eq. 1 where $\vec{\mathbf{i}}$ and $\vec{\mathbf{c}}$ mean the initial line vector and the correction vector, respectively. Figure 4(d) shows the baseline that is parallel to the real horizontal line. This process is indicated in Fig. 6.

$$\vec{\mathbf{b}} = \begin{cases} s \cdot \vec{\mathbf{i}} + \vec{\mathbf{c}} & (s = {}^{c}/_{i}, \text{if } c > i) \\ \vec{\mathbf{i}} + s \cdot \vec{\mathbf{c}} & (s = {}^{i}/_{\mathrm{c}}, \text{if } i > c) \end{cases} \tag{1}$$

3.2 Camera Parameter Estimation

After calculating the baseline, we estimate the rectified camera parameters. We firstly find the new camera centers. In the proposed method, the left camera center is considered as the reference and we estimate the new camera center of the right camera. Then the new camera center of the right camera is defined as a point that is apart with the user-input camera distance along the direction of the baseline.

After that, we consider the camera rotation matrices. We estimate each camera rotation matrix that satisfies the following conditions. The horizontal axis of every image plane becomes parallel to the baseline vector. All the principal axes are defined in common as the direction perpendicular to both of the baseline vector and the average of all the original vertical axes. Then, the vertical axis of each image plane is orthogonal to both of the new principal axis and the baseline vector. Thus, the rotation matrix for the rectified stereo camera **R**' has the form shown in Eq. 2, where $\vec{\mathbf{b}}$ and $\overrightarrow{\mathbf{v}_a}$ mean the directions of the baseline vector and the average of all the original vertical axes, respectively.

$$\mathbf{R}' = \begin{bmatrix} \vec{\mathbf{b}}^{\mathrm{T}} \\ ((\vec{\mathbf{b}} \times \overrightarrow{\mathbf{v}_a}) \times \vec{\mathbf{b}})^{\mathrm{T}} \\ (\vec{\mathbf{b}} \times \overrightarrow{\mathbf{v}_a})^{\mathrm{T}} \end{bmatrix} \tag{2}$$

Then, we estimate the common camera intrinsic parameters. The focal length and the principal point are obtained as the averages of their original values, respectively. The same focal length of each camera makes all image planes coplanar. There are also uniform horizontal displacement between corresponding points and few vertical mismatches in pixels between corresponding points due to the same principal point of each camera. Finally, we obtain the rectified camera projection matrices which are composed of the estimated camera parameters like Eq. 3.

$$\mathbf{P}'_k = \mathbf{A}'[\mathbf{R}' \,|\mathbf{t}'_k] = \mathbf{A}'[\mathbf{R}' \,|-\mathbf{R}' \,\mathbf{C}'_k] \tag{3}$$

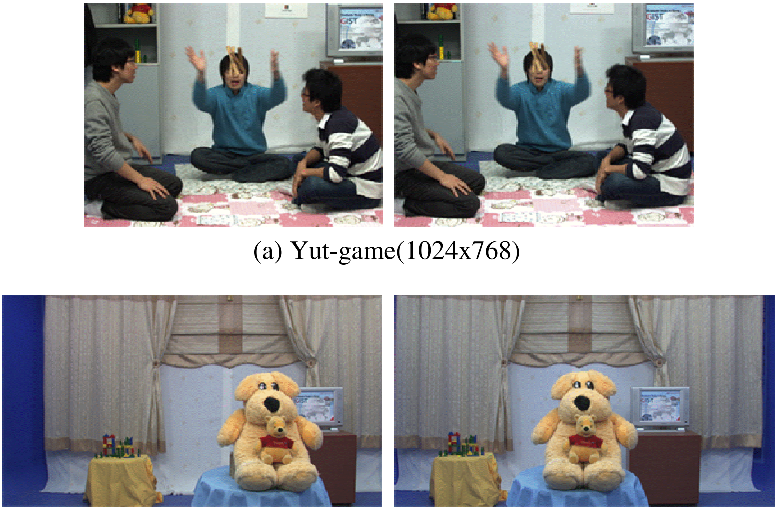
3.3 Rectification Transform

For the last step, we can generate the rectified stereo image by calculating and applying the rectification transform. We consider the epipolar geometry for each viewpoint. Then, we use the point-to-point mapping between images of the original and estimated cameras called the 2-D homography $\mathbf{H}$ [10]. Finally, the transform for k-th image is obtained by using this homography like Eq. 4. By applying this transform to each image, we can obtain the rectified stereo image.

$$\mathbf{T}_k = \mathbf{H}_{\pi,k} = \mathbf{P}'_k \mathbf{P}_k^{+} \tag{4}$$

4 Experimental Results

For experiments, we captured two sets of stereo image. We used two types of cameras; one provides 1024x768 and the other provides full HD (1920x1080) resolution. The first test images are shown in Fig. 7(a), the distance between two cameras is about 6.5cm. The second test images are shown in Fig. 7(b), the distance between two cameras is 40cm in this case. Figure 8(a) and Fig. 8(b) show that the captured images have the practical stereo geometry which is shown in Fig. 1. There were vertical pixel displacement between corresponding points, and also we notice the camera rotation difference.

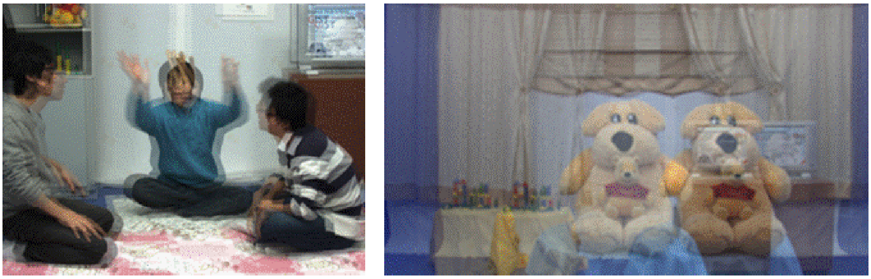


(a) Yut-game(1024x768)

(b) Bear(1920x1080)

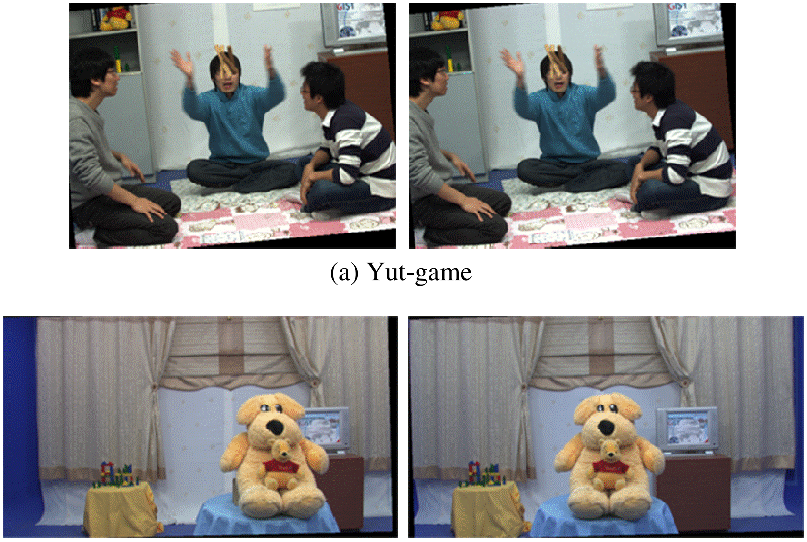
Fig. 7. Captured images

Figure 9 shows the rectified result by the conventional method [6]. We notice that the rectified images are skewed. This skewness is due to the skewed baseline connecting the original $\mathbf{C}_1$ and $\mathbf{C}_2$. Especially for the second image set, although there is a little geometrical misalignment in the original images, the rectified results have more visual distortion.



(a) Synthetic image of Yut-game (b) Synthetic image of Bear

Fig. 8. Synthetic images for the captured stereo images

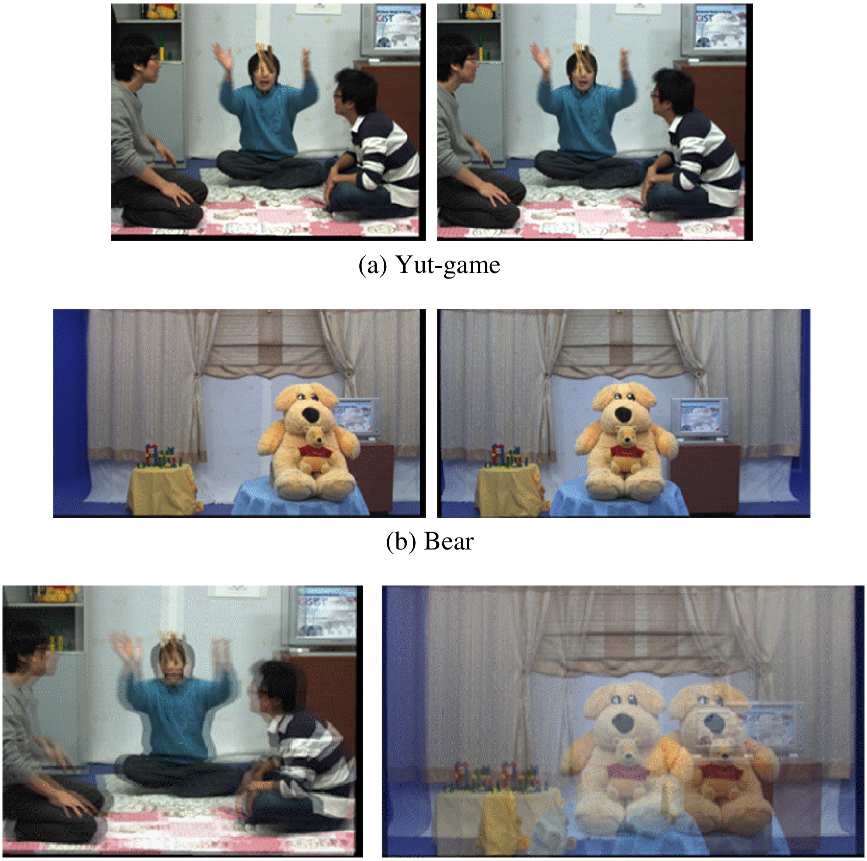


(a) Yut-game

(b) Bear

Fig. 9. Skewed result images

Figure 10 shows the rectified stereo images and their synthetic images by the proposed method. As shown in Fig. 10, the result images are not only rectified well but also almost parallel to the horizontal line in the real world. It is because the baseline for rectification is calculated to be parallel to the real horizontal line. Also, as shown in Fig. 10(c) and Fig. 10(d), there are few pixels of vertical difference between corresponding pixels.



(a) Yut-game

(b) Bear

(c) Synthetic image of Yut-game

(d) Synthetic image of Bear

Fig. 10. Results by the proposed method

5 Conclusion

In this paper, we presented a stereo image rectification method using the horizontal baseline. The proposed method avoids that the rectified images become skewed due to the miscalculated baseline. The baseline in the proposed method is calculated to be parallel to the horizontal line in the real world using the initial and correction vectors calculation. Therefore, the experimental results show that the results from the proposed method have less geometrical misalignment without the visual distortion compared to the conventional method.

Acknowledgements. This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2011-(C1090-1111-0003)).

References

1. Smolic, A., Kauff, P.: Interactive 3D Video Representation and Coding Technologies. Proc. of IEEE, Spatial Issue on Advances in Video Coding and Delivery 93(1), 99–110 (2005)
2. Sun, J., Zheng, N.N., Shum, H.Y.: Stereo Matching Using Belief Propagation. IEEE Transactions on Pattern Analysis and Machine Analysis (PAMI) 25(5), 787–800 (2003)
3. ISO/IEC JTC1/SC29/WG11 M12030: Comments on Input and Output Format of MVC (2005)
4. Hartley, R.: Theory and Practice of Projective Rectification. International Journal of Computer Vision 35(2), 115–127 (1999)
5. Loop, C., Zhang, Z.: Computing Rectifying Homographies for Stereo Vision. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 125–131 (1999)
6. Fusiello, A., Trucco, E., Verri, A.: A Compact Algorithm for Rectification of Stereo Pairs. Machine Vision and Application 12(1), 16–22 (2000)
7. Kang, Y., Lee, C., Ho, Y.: An Efficient Rectification Algorithm for Multi-view Images in Parallel Camera Array. In: Proc. of 3DTV Conference 2008, pp. 61–64 (2008)
8. Kang, Y., Ho, Y.: Geometrical Compensation for Multi-view Video in Multiple Camera Array. In: Proc. of International Symposium ELMAR, pp. 83–86 (2008)
9. Camera Calibration Toolbox for Matlab, `http://www.vision.caltech.edu/bouguetj`
10. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2003)

Real-Time Image Mosaicing Using Non-rigid Registration

Rafael Henrique Castanheira de Souza, Masatoshi Okutomi, and Akihiko Torii

Tokyo Institute of Technology
rafaelh.souza@ok.ctrl.titech.ac.jp , {mxo,torii}@ctrl.titech.ac.jp
http://www.ok.ctrl.titech.ac.jp

Abstract. Mosaicing is a classical application of image registration where images from the same scene are stitched together to generate a larger seamless image. This paper presents a real-time incremental mosaicing method that generates 2D mosaics by stitching video key-frames as soon as they are detected. The contributions are three-fold: (1) we propose a "fast" key-frame selection procedure based solely on the distribution of the distance of matched feature descriptors. This procedure automatically selects key-frames that are used to expand the mosaics while achieving real-time performance; (2) we register key-frame images by using a non-rigid deformation model in order to "smoothly" stitch images when scene transformations can not be expressed by homography: (3) we add a new constraint on the non-rigid deformation model that penalizes over-deformation in order to create "visually natural" mosaics. The performance of the proposed method was validated by experiments in non-controlled conditions and by comparison with the state-of-the-art method.

Keywords: mosaic, non-rigid, registration, feature based, real-time.

1 Introduction

Mosaicing is a classical application of image registration. Typically, a set of images is stitched together to simulate a camera with a larger field of view. Real-time mosaicing can be useful for medical imaging, augmented reality, digital camera panorama generation, etc. Online registration, i.e., stitching key-frames as soon as they are detected, is necessary for real-time processing.

In this work, we propose a method of online mosaicing that can generate 2D mosaics from video inputs acquired beyond homography assumptions. Classical mosaicing methods work under the assumption that the input images are related to each other by homography (*projective transformation*). This assumption holds true when the images are acquired under some limited conditions (camera rotatation around its optical center or scene lying on a planar surface). Unless these conditions are satisfied, the images can not be perfectly aligned by registration and the results may be very poor. This problem may be alleviated by the application of non-rigid registration [5].

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 311–322, 2011.

A naive approach to online mosaicing is to register and stitch the current key-frame into the previous key-frame. The process will accumulate registration error, which will grow with each new image added to the sequence.

This paper presents a method which uses a very efficient feature based non-rigid registration model in order to align images with high precision. At the same time, the over-deformation of the mosaic is avoided during the online mosaic creation. These two objectives are achieved by formulating the registration problem by enforcing smoothness while keeping the original proportions of the captured frame. Additionally, in order to achieve real-time processing, the key-frames are efficiently extracted from the video by a procedure which uses the distance distribution of matched feature descriptors.

In Sect. 2, the related methods are presented. Section 3 presents the proposed method. Section 4 shows the result of experimental validations. Finally, Sect. 5 shows the conclusions of this research and future research subjects.

2 Related Work

For the reader who is not familiar with mosaicing, Szeliski [13] presents a comprehensive tutorial about a variety of methods of registration and mosaic composition.

Since mosaic is a well studied area of computer vision, there are many approaches to 2D mosaicing. These works can be grouped in 3 classes: (1) offline methods that use homography or lower degree transformations, (2) offline methods that use higher degree transformations, and (3) online methods. The group (1) includes the works, [2,9,12,7], which are based on global transformations such as homography. The group (2) includes the works [4,3], which model the deformation as quadratic functions. The group (3), which is the most related to the proposed method, includes the works of [6,10]. The work in [6] uses 3D information for registering aerial images using a non real-time algorithm. The method in [10] is online and avoids the problem of over deformation by using fixed camera movements (*e.g.* translation, forward motion).

Although most of the presented works dealing with mosaicing make use of global transformations such as homography, there are more general registration methods that use non-rigid deformation. Some of them use feature based methods, *e.g.* [5,11,14]. Feature based methods are generally more computationally efficient than area based methods [13], specially in the case of non-rigid registration. The method in [14] can register correctly pairs of images even in the presence of a large ratio of outliers in real-time. However, this method is designed for pairs of images only.

On top of the state of the art, the contributions of the proposed method are: real-time performance, use of non-rigid registration, prevention of over-deformation of the mosaic, and less restrictions on camera movement.

3 Proposed Method

The mosaicing procedure consists of four steps: frame selection, feature matching, registration, and mosaic displaying. The frame selection module reads the input video and selects which key-frames will be used to create the mosaic. The feature matching module matches the feature points in the newly selected frame to the features in the previously selected frame. The pairwise registration module receives the set of matched features and registers the newly selected frame into the previously selected frame. The registered frame is then sent to the mosaic creation module, where it is added to the mosaic and displayed. The procedure is repeated again, until the end of the video. The modules are explained in more details in the following sections.

3.1 Frame Selection

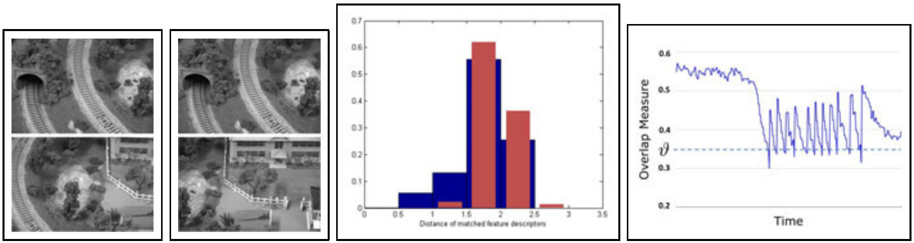
In order to create mosaics efficiently, only a small subset of the video frames must be selected. This key-frame set must be as sparse as possible, to reduce the number of registrations performed. At the same time, it must contain overlapping key-frames so that a mosaic can be composed out of them. To fulfill these requirements, it is necessary to estimate the overlap of pairs of frames. To do so, the following algorithm is proposed: (1) the features in both frames being compared are detected using SURF descriptors [1]); (2) the nearest-neighbor matching of the features is computed; (3) a histogram of the matched descriptors is computed; (4) the overlap measure (OM) is computed. The OM was defined as follows:

$$OM(H) = \sum_{j=1}^{n_{Bin}} G\left((j-0.5)h_{size}, \varsigma\right) H_j \quad , \tag{1}$$

where n_{Bin} is the number of bins in the histogram, h_{size} is the size of each bin, $(j-0.5)\,h_{size}$ is the average range of the bin j, G is a Gaussian weighting function with 0 mean and standard deviation ς. This weighting function assigns larger weights to values near zero, and the weight decays quickly, so that the bins which probably contain correct matches receive a larger weight than the bins with wrong matches. So, using the OM, the key-frames are selected by the following algorithm: (1) the first video frame is selected and used as reference; (2) the next frame whose OM (comparing with the reference frame) is smaller than a given threshold is selected and becomes the new reference. Step (2) is repeated until the end of the video.

It was experimentally observed that the probability distribution of the descriptor distances changes according to the intersection size between the image pair. Fig. 1(a) shows two frames with a small overlap. The descriptor distance has a bell-shape like distribution (fig. 1(b)). Fig. 1(c) shows two frames with a larger overlap. The distribution becomes bimodal (fig. 1(d)). The smaller peak represents the inliers among the matched features. Fig. 1 shows the variation of

OM over time, in a video recorded by a translating camera. The value of OM decreases as the intersection becomes smaller and rises again when a new frame is selected.



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

Fig. 1. (a): pair of frames with a large overlap; (b): pair of frames with a small overlap; (c): histogram of the distance of matched descriptors; the blue bars represent pair (a) and the red bars the pair (b); (d): variation of the overlap measure over time

3.2 Registration

This section explains the registration model used in the proposed method. Two constraints must be met: the mosaic must be as seamless as possible and as similar as possible to the original captured frame (i. e., over-deformation must be avoided). For doing so, the proposed method applies a non-rigid deformation model that uses triangle meshes and a registration algorithm that uses feature points obtained by the frame selection procedure and pruned by RANSAC [8].

Deformation Model for Image Registration. A 2D mesh model is used to implement the non-rigid transformations. Each vertex (or control point) v_j is represented by its coordinates (x_j, y_j). The entire mesh is written as $S = (X, Y)$, where X is a vector containing the x coordinates of the control points and Y the vector containing the y coordinates. The warp of any point p inside a mesh triangle defined by the vertices v_i, v_j, and v_k can be calculated using the barycentric coordinates of p: $w(p, S) = \sum_{l \in \{i,j,k\}} B(p, v_l) [x_l, y_l]^{\mathrm{T}}$, where $B(p, v_l)$ is the barycentric coordinate of p in relation to $v_l \in \{v_i, v_j, v_k\}$ (computed in relation to the identity mesh S_0). Fig. 2 illustrates the basic principle of this kind of transformation.

Problem Formulation. The initial model of pairwise non-rigid registration was drawn from Zhu *et al.*'s work [14], which was based on Pilet *et al.*'s work [11]. It is summarized by the equation below:

$$E(S) = E_C(S) + \lambda E_{\mathrm{Sm}}(S) \quad , \tag{2}$$

where E_C is the correspondence energy function and E_{Sm} is the smoothness energy. The constant λ balances the compromise between precision and mesh smoothness. The registration is solved by finding the mesh S which minimizes

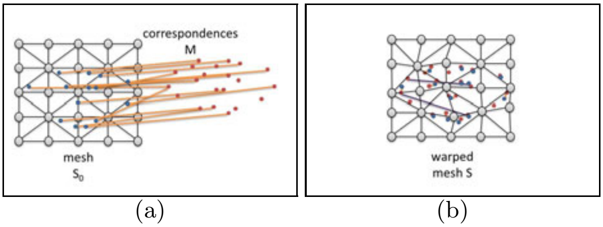


Fig. 2. Deformation using a mesh model: (a) shows the identity mesh, (b) shows the mesh S warped to reduce the projection error of the matched features

$E(S)$. The correspondence energy is proportional to the projection error of warped features, while the smoothness energy measures the discontinuities on S; this energy is important to remove outlier feature matchings. The initial formulation described by (2) is suitable for pairwise image registration, however. The registration of sequences of images poses some additional problems. If only pairwise registration is used to align a sequence of images, over-deformation due to error accumulation may occur (fig. 3).

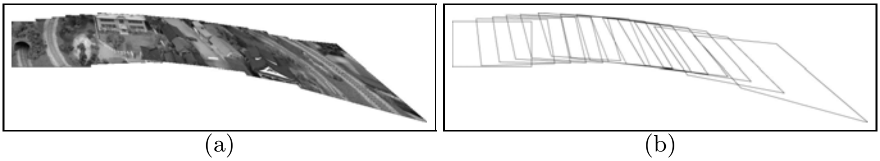


Fig. 3. Error accumulation using homography: (a) rendered mosaic , (b) projected frame borders. The last frame is the most deformed.

To avoid error accumulation, a modified version of the previous energy function is presented. The new term, $E_{\mathrm{Ref}}(S - S_{\mathrm{Ref}})$ is named *reference mesh energy.* The mesh S_{Ref} represents a model of how the mesh S should look like without over-deformation. Alternatively, it is how the user of the mosaic system would expect the image (warped by S) to look like. The constant μ regulates the reference mesh energy weight. The new formulation is presented below:

$$E'(S) = E_C(S) + \lambda E_{\mathrm{Sm}}(S) + \mu E_{\mathrm{Ref}}\left(S - S_{\mathrm{Ref}}\right) \ . \tag{3}$$

Correspondence Energy. The correspondence energy $E_{\mathrm{C}}(S)$ is a function of the projection error of the matched features. The matched feature set is represented by M. The matched feature pair $c \in M$ is composed of two features (c_0, c_1), where c_0 is a feature found in the target image and c_1 is its paired feature found in the image being warped. The warp function $w(c_1, S)$. The function υ is the same robust estimator used by Zhu *et al.*[14]. It is defined below:

$$E_{\mathrm{C}}(S) = \sum_{c \in M} \upsilon\left(c_0 - w\left(c_1, S\right), \sigma\right); \upsilon(\delta, \sigma) = \begin{cases} \frac{||\delta||^2}{\sigma^n} & \text{if } ||\delta|| \leq \sigma \\ \sigma^{2-n} & \text{otherwise} \end{cases} \tag{4}$$

The function υ has two parameters: the projection error δ and the radius of tolerance σ. The matches whose projection error are greater than the radius of tolerance are considered outliers and penalized. The radius of tolerance σ dictates which matched feature pairs will be considered outliers, conferring robustness to the registration procedure.

Smoothness Energy. The correspondence energy, if used alone, is sensitive to outliers among the matched features. A smoothness constraint is added to the model in order to avoid this problem. The proposed method uses the same smoothness constraint found in Zhu *et al.*[14] and Pilet *et al.*[11]. This energy is the sum of the approximate second derivative of the mesh S. Let E be the set of all collinear control points in S that define two adjacent edges. The smoothness energy is defined below:

$$E_{\mathrm{Sm}}(S) = \sum_{i,j,k \in E} (-x_i + 2x_j - x_k)^2 + (-y_i + 2y_j - y_k)^2 = X^{\mathrm{T}}KX + Y^{\mathrm{T}}KY \ , \tag{5}$$

where $K = K'^{\mathrm{T}}K'$, and K' is a matrix containing one row per triplet in E and one column per mesh vertex. The row corresponding to the triplet (i, j, k) has all of its values zero except by values in columns i, j, and k, that have values -1, 2, and -1, respectively [11].

Reference Mesh Energy. The registration using the energy function in (2) is only suited for pairwise registration, because registration error may accumulate, as shown in fig. 3. The role of the reference mesh energy is to alleviate this problem. This energy is proportional to the L_2 distance between the mesh S and the reference mesh S_{Ref}. The former is the registration solution and the latter is an approximation of how S should be if it has no over-deformation. The criteria selected to generate S_{Ref} was to make S_{Ref} look similar to the original captured image. S_{Ref} is defined as the similarity transformation (i.e., rotation, translation and scaling) that minimizes the correspondence energy. This mesh can be computed efficiently by reducing the projection error using the similarity transformations combined with RANSAC. The reference mesh energy is defined below:

$$E_{\mathrm{Ref}}\left(S - S_{\mathrm{Ref}}\right) = \frac{1}{2}||S - S_{\mathrm{Ref}}||^2 \ . \tag{6}$$

During the optimization process, the reference mesh energy is stronger in the regions of the mesh S where there are no features. While the region with features is deformed to minimize the projection error, the region without features is deformed by similarity transformations. These local differences in the deformation are not possible for global registration models.

Optimization Routine. As pointed in [14], the projection error δ can be written as a linear system. Given that: $c_0 = (c_{0x}, c_{0y})$, $c_1 = (c_{1x}, c_{1y})$:

$$||\delta||^2 = \left(c_{0x} - t^{\mathrm{T}}x\right)^2 + \left(c_{0y} - t^{\mathrm{T}}y\right)^2 \ , \tag{7}$$

where x and y are the coordinates of the mesh and $t_{c_1} \in R^N$ is a vector (N is the number of control points) representing the barycentric coordinates of the feature point c_1, which is inside the triangle defined by $v_i, v_j, v_k \in S_0$, (calculated in the identity mesh). The vector t_{c_1} has all its values 0, except in the coordinates i, j, and k, where the barycentric coordinates of c_1 in relation to v_i, v_j, and v_k are set, respectively. Using (5) and (6), the energy $E'(S)$ in (3) can be rewritten as:

$$E'(S) = \frac{1}{\sigma^n} \sum_{c \in M_{Inl}} \left(c_{0x}^2 + c_{0y}^2 - 2 \begin{bmatrix} c_{0x} t \\ c_{0y} t \end{bmatrix}^{\mathrm{T}} S + S^{\mathrm{T}} \begin{bmatrix} tt^{\mathrm{T}} \; 0 \\ 0 \; tt^{\mathrm{T}} \end{bmatrix} S \right) + \\ |M_{Out}| \sigma^{2-n} + \lambda \left(X^{\mathrm{T}} K X + Y^{\mathrm{T}} K Y \right) + \frac{\mu}{2} ||S - S_{\mathrm{Ref}}||^2 \;, \tag{8}$$

where M_{Inl} is the set of inlier matches, M_{Out} is the set of outlier matches. The following definitions are done for simplification: $A = \frac{1}{\sigma^n} \sum_{c \in M_{Inl}} tt^{\mathrm{T}}$, and $b = \begin{bmatrix} b_x \\ b_y \end{bmatrix} = \frac{1}{\sigma^n} \sum_{c \in M_{Inl}} \begin{bmatrix} c_{0x} t \\ c_{0y} t \end{bmatrix}$. Computing the gradient of E' and setting it to zero, the mesh S can be found by solving a linear system:

$$S = \begin{bmatrix} \lambda K + A + \mu I & 0 \\ 0 & \lambda K + A + \mu I \end{bmatrix}^{-1} \left(b + \mu S_{\mathrm{Ref}} \right) \;. \tag{9}$$

The optimization is repeated varying the value of σ, which decreases during the optimization procedure. In the beginning, σ is large, allowing many possible outliers to influence the result of the optimization process. However, since the module of the derivative of the E_{C} is small when σ is large, E_{Sm} and E_{Ref} have a larger weight and they initially guide the optimization. As the value of σ decreases, the weight ofE_{C} increases, guiding the optimization to minimize the projection error of the remaining inliers. In this way, this registration method is robust to outliers. The process stops when σ is smaller than a given threshold.

In order to display the results, the mosaics are created by warping the registered frames one over the other. In order to avoid using regions without features, that may have large registration errors, only the convex hull of the correctly aligned feature points is warped.

4 Experiments

The objective of the experiments is to demonstrate four points: the proposed method has a smaller projection error comparing to the classical approaches, the mosaics created by the proposed method have less over-deformation, the proposed method can run in real-time, and that the results obtained by the proposed method are more robust than the results obtained by classical approaches in the kind of video considered.

4.1 Experimental Setup

The project was run in a computer with Intel(R) Core(TM) i7 CPU (2.93 GHz) and 4GB of RAM. The proposed method was implemented using the OpenCV library. The parameter setting is presented in Table 1.

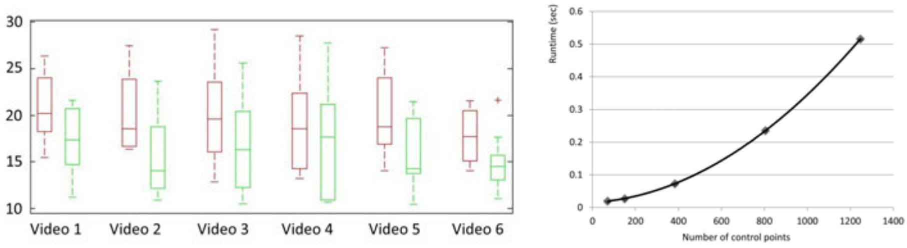
Table 1. Parameter settings for the proposed method

Param.	Value	Description
ϑ	0.4	Frame selection threshold.
ς	1.0	Frame selection weight function std. deviation.
λ	10^{-6}	Smoothness energy parameter.
μ	10^{-7}	Reference mesh energy parameter.
n	4	Correspondence energy parameter.
σ_0	32	Registration parameter; initial radius of tolerance.
σ_{min}	3	Minimum radius of tolerance; i.e., projection error.
η	0.5	Radius of tolerance decay rate.

For the reference mesh computation, the precision of RANSAC is set to 99% in the presence of 70% of outliers. The size of the mesh was 19 × 28 control points. The videos used on the experiments had a resolution of 720 × 480.

4.2 Registration Precision

This experiment presents the comparison between homography and non-rigid transformations concerning precision by means of mean appearance error, defined as the mean absolute difference between between all aligned pixels. The experiments were conducted by registering of pairs of images. Fig. 4(a) shows the results of the average error of pair-wise registration over different video sequences. Fig. 5 shows a detail of a pair of registered frames (the averaged image). As can be seen, the results achieved by the registration method used by the proposed method are always more precise than the results using homography. This happens because the deformation field between the pairs of images can not be precisely described by a global transformation like projection, since the displacement field depends on the geometry of the scene.



(a) (b)

Fig. 4. (a): appearance error with homography and non-rigid transformations. The error is measured as the mean absolute difference between pixel gray-scale values of aligned pixels, in a set of videos. The red boxes show the results obtained by homography, and the green boxes represent the results of the proposed method; (b): execution time (seconds) in relation to number of control points.

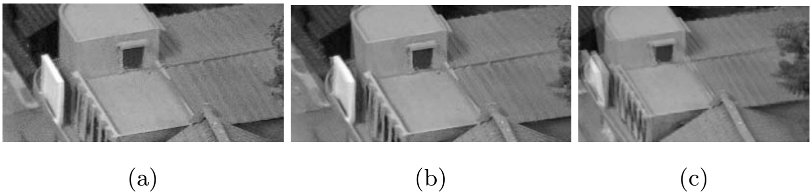


Fig. 5. Detail of a pair of registered frames, showing the average of the superposition of the frames, aligned by: (a) shows the original frame from the video, (b) proposed method and (c) homography

4.3 Over-Deformation Avoidance

This set of experiments compares mosaics done by the proposed method and non-rigid registration as described by [14]. The comparisons are done regarding over-deformation. Figure 6 shows the results. Both methods use the same set of frames. As previously showed in fig. 3, using homography, the registration error tends to build up and cause the frames to over-deform. When using only non-rigid registration, without the reference mesh energy, error accumulation also happens, even though the alignment error is small when compared to homography. The proposed method, using the reference mesh energy, minimizes these amount of over-deformation. This result may be achieved by related methods using bundle adjustment, but the proposed method achieves the same by only doing pair-wise registration.

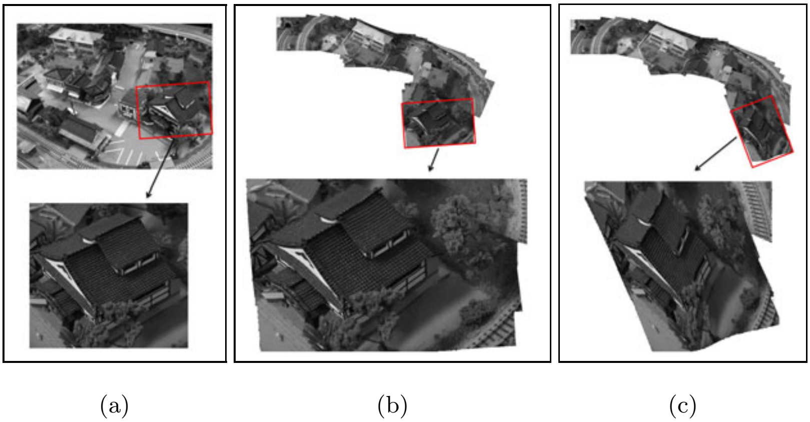


Fig. 6. Mosaicing results, regarding overdeformation; (a) detail of the city model used in the experiments, showing an expected undeformed frame; (b) shows the result obtained by the proposed method; and (c) shows the results obtained using only non-rigid registration without the reference mesh energy

4.4 Comparison with a Standard Method

In this set of experiments, the proposed method was compared to a standard method, implemented by Microsoft Image Composite Editor (ICE), version 1.3.5. Using ICE, the user can choose different camera movements. The one which yielded the best result was selected. The proposed method used the parameters described in section 4.1. ICE and the proposed method used the same set of key-frames. Fig. 7 shows the mosaic created from a video taken by a camera moving over a city model. The results can be seen below.

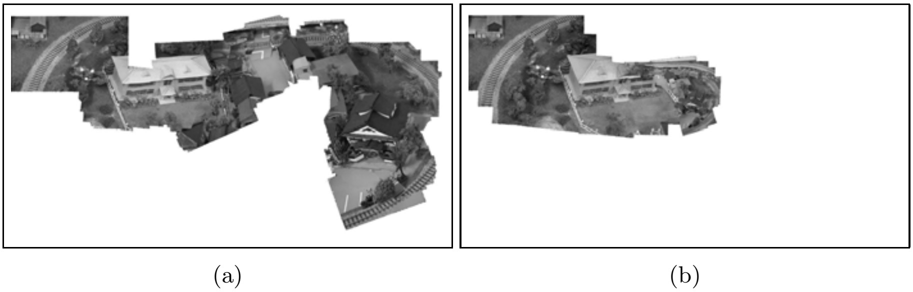


Fig. 7. Comparison between the proposed method and a standard method; (a) shows the result of the proposed method; (b) shows the result of the standard method. The proposed method created a more complete mosaic since it can handle more complex camera movements.

Fig. 7(a) shows the results obtained by the proposed method. Fig. 7(b) shows the results obtained by ICE. As can be seen, the results obtained by the proposed method are more complete than the results given by ICE. This happens because of the complex camera movement and the non-planar surface, which violate the projection constraints used by ICE.

4.5 Computational Complexity

The current implementation of the proposed method runs in about 32 frames per second with a tax of of 2 frames selected per second, what is reasonable for videos where the camera movement is not excessively fast.

Each iteration of frame selection takes approximately 0.031 seconds, so the frame rate is about 32 frames per second, enough for most videos. Fig. 4(b) shows runtime regarding only the registration procedure. It was executed 10 times for each quantity of control points (the computation of the reference mesh is included). As can be seen from the experiments, registration runtime grows slowly. This happens because the implementation that uses sparse matrices to represent the registration model. The runtime of the frame selection and mosaic creation procedure were also computed. Using approximately 1000 triangles, the registration can be done in about 3 frames per second. Regarding the mosaic creation, each frame takes on average 0.4 seconds to be added into the mosaic, a tax of nearly 2 frames per second.

The conclusion is that the proposed method can run in real time, given the conditions above. Further optimization on the method may be performed in the future.

5 Conclusions and Future Work

This paper presented a new mosaicing technique based on feature based non-rigid registration. The proposed method can be used to create mosaics of non-planar surfaces in real-time. This model deals with the problem of over-deformation using only pairwise registration, and creates mosaics with smaller alignment error when comparing with standard approaches. For this purpose, the reference mesh energy was presented. An efficient method of key-frame selection, created to achieve real-time performance, was also presented. The proposed method has some restrictions. First, since there is no bundle adjustment, the generated mosaic is prone to error if a region of the scene is recorded twice (loop). This will require efficient loop closing method able to run in real-time. The proposed method also fails when sharp discontinuities in the optical flow are present, due to the smoothness constraint. These limitations will be tackled in our future research.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
2. Brown, M., Lowe, D.: Automatic panoramic image stitching using invariant features. International Journal of Computer Vision 74, 59–73 (2007)
3. Can, A., Stewart, C.V., Roysam, B., Tanenbaum, H.L.: A feature-based technique for joint, linear estimation of high-order image-to-mosaic transformations: application to mosaicing the curved human retina. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 585–591 (2000)
4. Chaiyasarn, K., Kim, T.-K., Viola, F., Cipolla, R., Soga, K.: Image mosaicing via quadric surface estimation with priors for tunnel inspection. In: 2009 16th IEEE International Conference on Image Processing (ICIP), pp. 537–540 (2009)
5. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding 89(2-3), 114–141 (2003)
6. Crispell, D., Mundy, J., Taubin, G.: Parallax-free registration of aerial video. In: Proc. British Machine Vision Conf. (2008)
7. Deng, Y., Zhang, T.: Generating panorama photos. In: Proc. of SPIE Internet Multimedia Management Systems IV (2003)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 381–395 (1981)
9. Hsu, S., Sawhney, H.S., Kumar, R.: Automated mosaics via topology inference. IEEE Computer Graphics and Applications 22(2), 44–54 (2002)

10. Peleg, S., Rousso, B., Rav-Acha, A., Zomet, A.: Mosaicing on adaptive manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1144–1154 (2000)
11. Pilet, J., Lepetit, V., Fua, P.: Real-time non-rigid surface detection. In: Proc. IEEE Conf. Computer Vision Pattern Recognition, pp. 822–828 (2005)
12. Sawhney, H.S., Hsu, S., Kumar, R.: Robust Video Mosaicing Through Topology Inference and Local to Global Alignment. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 103–119. Springer, Heidelberg (1998)
13. Szeliski, R.: Image alignment and stitching: a tutorial. Found. Trends. Comput. Graph. Vis. 2, 1–104 (2006)
14. Zhu, J., Lyu, M.R., Huang, T.S.: A fast 2d shape recovery approach by fusing features and appearance. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 1210–1224 (2009)

Adaptive Guided Image Filtering for Sharpness Enhancement and Noise Reduction

Cuong Cao Pham, Synh Viet Uyen Ha, and Jae Wook Jeon*

School of Information and Communication Engineering,
Sungkyunkwan University, Suwon, South Korea
cuongpc@skku.edu, synhha@ece.skku.edu, jwjeon@yurim.skku.ac.kr
http://micro.skku.ac.kr

Abstract. Sharpness enhancement and noise reduction play crucial roles in computer vision and image processing. The problem is to enhance the appearance and reduce the noise of the digital images without causing halo artifacts. In this paper, we propose an adaptive guided image filtering (AGF) able to perform halo-free edge slope enhancement and noise reduction simulaneously. The proposed method is developed based on guided image filtering (GIF) and the shift-variant technique, part of adaptive bilateral filtering (ABF). Experiments showed the results produced from our method are superior to those produced from unsharp masking-based techniques and comparable to ABF filtered output. Our proposed AGF outperforms ABF in terms of computational complexity. It is implemented using a fast and exact linear-time algorithm.

Keywords: Edge-preserving smoothing, guided image filter, sharpness enhancement, noise reduction.

1 Introduction

Enhancing the sharpness and reducing the noise of the digital images have attracted much interest during the last decades. These pre-processing techniques play crucial roles in computer vision and image processing. However, how to simultaneously reduce noise and increase the slope of edges without creating halo artifacts is still a challenging issue.

Conventional linear filter effectively smooths noise in homogeneous regions, however, blurring the edges of an image. Conversely, edge-preserving smoothing techniques only filter noise, while preserving edge structures. Existing techniques that feasibly perform this kind of operation include anisotropic diffusion (AD) [14], bilateral filtering (BLF) [17] and guided image filtering (GIF) [9]. However,

* This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)" (NIPA-2011-(C1090-1121-0008)), and by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology(2011-0018397).

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 323–334, 2011.

none of them can be directly applied to achieve sharpness enhancement and noise reduction simultanenously, as is our stated goal.

Anisotropic diffusion is able to preserve and sharpen edges, but both noise and fine details are unexpectedly removed due to its over-smooth characteristic. Although BLF is widely used and has become the *de facto* standard for computer vision and image processing, its ability to enhance the sharpness of an image is limited. While GIF proposed in [9] outperforms BLF in a variety of computer vision applications, it shares the same limitation as does BLF.

In terms of image sharpening, the unsharp masking technique (USM) is popularly used due to its simplicity. A high-pass filter (HPF) is applied to the input image under the guidance of the unsharp mask, obtained by subtracting the input image and its blurred version. Thus, the contrast along the edges is increased in the sharpened output. However, as discussed in [3,10,19], USM has two major drawbacks. First, the overshoot and undershoot artifacts occur around the edges of the sharpened image due to the large boost of high contrast areas. Second, HPF not only enhances the edges but also significantly amplifies the noise in the input image. This reduces image quality.

Investigations have been conducted to improve these two limitations of USM [3,10,15]. Especially, Kim et al. proposed the optimal unsharp mask (OUM) [10] to reduce noise in the homogeneous regions, while achieving the equivalent level of sharpness as USM does. The Laplacian of Gaussian (LoG) filter is used to determine the locally adaptive optimal λ value, instead of a fixed λ for HPF. However, the halo artifacts have not been overcome completely.

In summary, state-of-the-art edge-preserving smoothing techniques cannot be used to achieve the goal directly, while unsharp masking-based approaches create overshoot and undershoot artifacts during the sharpening process. In a notable recent work, Zhang et al. [19] made use of the shift-variant technique to propose an adaptive bilateral filter (ABF) able to enhance the sharpness and remove the noise simultaneously. Unfortunately, the introduction of locally adaptive optimal parameters make this approach infeasible to fully adapt with the existing BLF acceleration schemes. It must be implemented using the two nested loops brute-force approach, whose computational complexity is $O(|w|^2)$, where $|w|$ is the size of the filter kernel.

In this paper, we present adaptive guided image filtering (AGF) for image sharpening and de-noising. Our proposed AGF method is based on GIF and the shifting technique proposed in [19]. The optimal training parameters produced from [19] are slightly modified and reused in our method to visually compare to ABF and OUM. However, we will prove the participation of these adaptive parameters does not corrupt the acceleration scheme of GIF - the O(N) time exact algorithm can still be applied to achieve the speed up. Experiments show the results produced from our method are superior to those produced from USM and OUM and comparable to ABF filtered results.

The remainder of this paper is organized as follows. Section 2 presents the connection between bilateral filter and guided image filter. Section 3 examines the adaptive bilateral filter with the shift-variant technique and adaptive optimal parameters. Section 4 presents the adaptive guided image filtering using the shift-variant technique. Section 5 presents the experimental results to compare our method to methods from the literature. Finally, this paper is drawn to a conclusion and future work outlined in Section 6.

2 Bilateral and Guided Image Filtering

In this section, we present the relationship between BLF and GIF in terms of the filter kernel. These two edge-preserving smoothing techniques play a central role in ABF and our proposed AGF.

2.1 Bilateral Filtering

As we briefly mentioned above, BLF is widely used due to its appealing characteristics. The name bilateral filter was first termed in [17] based on the work [1,16]. It is a non-iterative, non-linear filter that smooths low gradient regions, while preserving strong edges. Each output pixel is computed as a weighted mean of its neighbors. The weight is computed based on the spatial domain, like other linear filters, and on the intensity range domain. Let I_p be the intensity value at pixel p, w_k be the kernel window centered at pixel k, BLF is given by:

$$BLF(I)_p = \frac{1}{\sum\limits_{q \in w_k} W_{BLF_{pq}}(I)} \sum_{q \in w_k} W_{BLF_{pq}}(I) I_q \tag{1}$$

where the division term normalizes the weights sum to 1 and the kernel weights function $W_{BLF_{pq}}(I)$ can be expressed by:

$$W_{BLF_{pq}}(I) = \exp\left(-\frac{\|p-q\|^2}{2\sigma_s^2}\right) \exp\left(-\frac{|I_p - I_q|^2}{2\sigma_r^2}\right) \tag{2}$$

where the standard deviations parameters σ_s and σ_r control the decrement of weights in the spatial and intensity range domains, respectively. Each domain is represented by a Gaussian function. The spatial domain gives higher weight to pixels closer to the center pixel, whilst lower weight is assigned to distant pixels. Correspondingly, the same rule can be applied to the intensity range domain. Higher or lower weight will be assigned to the pixels that are similar to or different from the center pixel in terms of intensity value. The degree of smoothing can be adjusted by changing the value of σ_r. In most applications, this value must be sufficiently small to avoid filtering meaningful features, because BLF becomes equivalent to the Gaussian filter when σ_r increases.

Excessive time consumption is one of BLF's disadvantages, although it is efficient to implement. The brute-force approach consists of two nested loops. The computational complexity is $O(|w|^2)$, where $|w|$ is the size of the spatial domain. Studies have investigated reducing the time-taken [5,7,11,13]. The main concepts of these acceleration schemes can be found in [12]. Notably, the fast approximation approach proposed in [11] has been proved to be a very useful technique. It has been applied to a variety of bilateral-based applications [12].

2.2 Guided Image Filtering

He et al. [9] proposed GIF to overcome the gradient reversal artifacts occurring, using BLF in detail manipulation technique that is not mentioned in this paper. Instead, we focus on its ability of edge-preserving and fast implementation. It has been analyzed and proved that GIF shares the good edge-preserving characteristic compared to BLF. Furthermore, its fast and exact linear-time algorithm outperforms BLF in terms of computational complexity.

The filtering process of GIF is originally done under the guidance of an image G that can be another image or the input image I itself. It is similar to the joint bilateral filter [12] which is used to denoise the no-flash image I using the flash image G. When I and G are identical, joint bilateral filter becomes bilateral filter naturally. We first express GIF in terms of the filter kernel to establish the connection between BLF and GIF. Let I_p and G_p be the intensity value at pixel p of the input and guided image, w_k be the kernel window centered at pixel k, to be consistent with BLF. GIF is then formulated by:

$$GIF(I)_p = \frac{1}{\sum\limits_{q \in w_k} W_{GIF_{pq}}(G)} \sum_{q \in w_k} W_{GIF_{pq}}(G) I_q \tag{3}$$

where the kernel weights function $W_{GIF_{pq}}(G)$ can be expressed by:

$$W_{GIF_{pq}}(G) = \frac{1}{|w|^2} \sum_{k:(p,q) \in w_k} \left(1 + \frac{(G_p - \mu_k)(G_q - \mu_k)}{\sigma_k^2 + \varepsilon}\right) \tag{4}$$

where μ_k and σ_k^2 are the mean and variance of guided image G in local window w_k, $|w|$ is the number of pixels in this window. The key to understanding the edge-preserving ability of GIF lies in the term $1+[(G_p - \mu_k)(G_q - \mu_k)]/(\sigma_k^2 + \varepsilon)$ in this equation. When both G_p and G_q are concurrently on the same side of an edge (smaller or larger than the mean), the weight assigned to pixel q is large. Conversely, a small weight will be assigned to pixel q when they are on different sides (one is smaller and one is larger than the mean). Some further computations in [9] confirm the normalization term in equation (3) equals 1. The filter kernel of GIF can be shortened as follows:

$$GIF(I)_p = \sum_{q \in w_k} W_{GIF_{pq}}(G) I_q \tag{5}$$

The degree of smoothing of GIF is adjusted via parameter ε. The larger the value of ε is, the smoother the filtered image will be. It plays an equivalent role to σ_r in BLF. Some further experiment and demonstration in [9] prove that BLF and GIF yield approximately equivalent smoothing results, by setting $\varepsilon = \sigma_r^2$ in the normalized $[0; 1]$ intensity range value. Of course, the guided image G is identical to the input image I in this relation. This property is crucial, because it is going to be used to convert the optimal parameters of ABF to our proposed AGF, as shown in Section 4.

The $O(N)$ exact algorithm of GIF is performed by applying a chain of box filters using the $O(N)$ time integral image technique [6]. The linear translation-variant takes the place of the filter kernel (4) when computing this fast and exact linear-time algorithm. We will discuss this issue in more detail in Section 4.2.

3 Adaptive Bilateral Filtering

In this section, we examine ABF for sharpness enhancement and noise removal. We mainly focus on the shift-variant technique, because it will be applied to our method. This method was proposed in [19] based on the work [18]. The main differences of ABF compared to BLF is the introduction of the shifting technique and locally adaptive optimal parameters. These modifications make ABF outperforms conventional BLF in terms of image sharpening and de-noising. The filter kernel and weighting function of ABF are expressed by:

$$ABF(I)_p = \frac{1}{\sum\limits_{q \in w_k} W_{ABF_{pq}}(I)} \sum_{q \in w_k} W_{ABF_{pq}}(I) I_q \tag{6}$$

$$W_{ABF_{pq}}(I) = \exp\left(-\frac{\|p-q\|^2}{2\sigma_s^2}\right) \exp\left(-\frac{|(I_p+\xi_p)-I_q|^2}{2\sigma_r^2}\right) \tag{7}$$

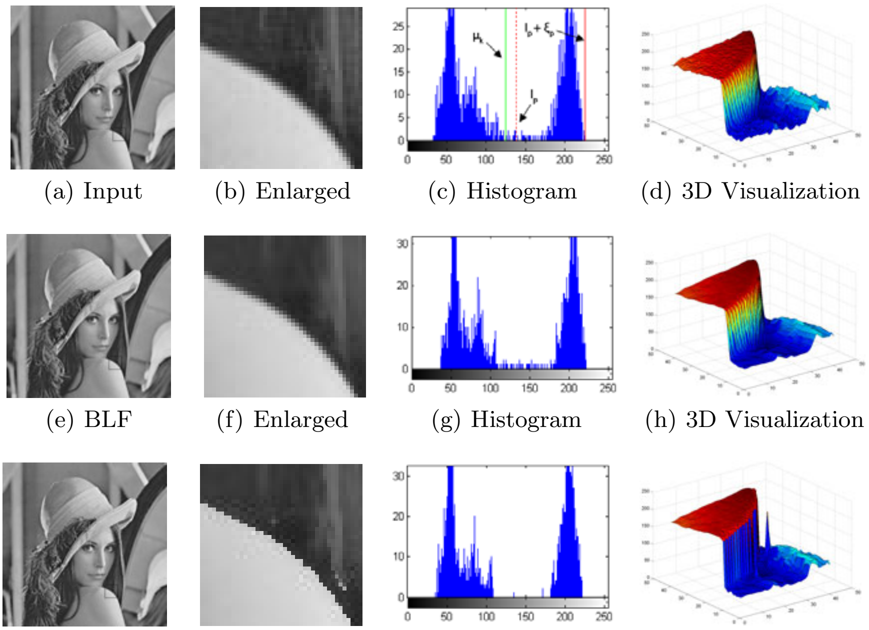
where ξ_p is the introduced offset that enables ABF to sharpen the image. The näive strategy for choosing this value is guided by:

$$\xi_p = \begin{cases} \mathrm{MAX}(w_k) - I_p & \text{if } \Delta_p > 0 \\ \mathrm{MIN}(w_k) - I_p & \text{if } \Delta_p < 0 \\ 0 & \text{if } \Delta_p = 0 \end{cases} \tag{8}$$

where $\Delta_p = I_p - \mu_k$ is the intensity difference between pixel p and the mean of local window w_k. While $\mathrm{MAX}(w_k)$ and $\mathrm{MIN}(w_k)$ are the maximum and minimum values of local window w_k, respectively.

This strategy is due to the histogram analysis, as shown in Fig. 1. For an input image shown in Fig. 1(a), the histogram and 3-D visualization of its enlarged window (Fig. 1(b)) are shown in Fig. 1(c) and 1(d), respectively. For the conventional BLF, the intensity range domain normally computes the affinities between the center pixel p and its neighbors q. This center value I_p is represented by the dotted-red line in the histogram. Thus, the slope of the edge in the filtered output is only just preserved, but not sharpened. The second row of Fig. 1 represents the corresponding conventional BLF output. In contrast, the edge is extremely enhanced by applying ABF with the näive offset choosing strategy. The center value I_p has been shifted to $\mathrm{MAX}(w_k)$ (red line), because its intensity value (dotted-red line) is larger than the mean μ_k (green line). However, as we can see in the third row, the aliasing effect and unexpected outliers occur in the sharpened output.

Zhang et al. proposed a more reliable strategy for choosing offset value to overcome this problem. They estimated both the offset ξ and standard deviation



(a) Input (b) Enlarged (c) Histogram (d) 3D Visualization

(e) BLF (f) Enlarged (g) Histogram (h) 3D Visualization

(i) Näive ABF (j) Enlarged (k) Histogram (l) 3D Visualization

Fig. 1. Illustration of the effect of ABF ($\sigma_s = 1.0, \sigma_r = 20$) with the näive offset choosing strategy compared to conventional BLF ($\sigma_s = 1.0, \sigma_r = 20$)

σ_r of the intensity range domain via a training procedure. Given the N sets of training images, where each set S consists of a high-quality original image I, a degraded image J and its restored output $\hat{J}$, the optimal parameters are obtained by solving the following minimum mean squared error estimation problem:

$$\{\xi_i^*, \sigma_{r,i}^*\} = \underset{\{\xi_i, \sigma_{r,i}\}}{\arg\min} \sum_{n=1}^{N} \left\| I_{n,p} - \hat{J}_{n,p} \right\|_{S^{(n)}}^2 \tag{9}$$

where $i = 1, 2, \ldots, T$ is the pixel classified number obtained by applying a 9×9 Laplacian of Gaussian filter (LoG) with $\sigma_{LoG} = 1.5$. The resultant parameters are locally adaptive, making ABF more robust. Zhang et al. [19] show how to find these parameters; we refer the reader to their paper for further details.

The main concern when applying ABF with optimal parameters is the large computational cost of its brute-force implementation. The standard deviation σ_r must be fixed in order to accelerate it using the method [11]. Otherwise, it will degrade the 3-D convolution model of method [11] when applying both adaptive offset and standard deviation parameters. Our proposed AGF with the use of these optimal parameters can achieve comparable results to ABF, while still keeping the generality of the linear translation-variant of the GIF. That is, the exact and linear-time algorithm is easily applied to achieve the acceleration.

4 Proposed Adaptive Guided Image Filtering for Image Sharpening and De-noising

4.1 Proposed Adaptive Guided Image Filtering

In this section, we present our proposed method using the shifting technique. As we have seen when we analyzed the relationship between BLF and GIF in terms of the filter kernel in Section 2, the main difference between them lies in their weighting functions of the filter kernel, as shown in equation (2) and (4). However, the intensity range domain of BLF and kernel function of GIF are similar in principle, because each of them takes the intensity value of center pixel p, local neighbors q and a smoothing parameter (σ_r in BLF, ε in GIF) in the computation process.

This is based on the shifting technique of ABF, in which the offset ξ_p is added to the intensity value of center pixel p in the intensity range domain of BLF. The same strategy is applied to our proposed AGF - the offset is added to the intensity value of center pixel p in the kernel weights function of GIF. Formally, the filter kernel and weighting function of our proposed AGF are given by:

$$AGF(I)_p = \sum_{q \in w_k} W_{AGF_{pq}}(G) I_q \tag{10}$$

$$W_{AGF_{pq}}(G) = \frac{1}{|w|^2} \sum_{k:(p,q) \in w_k} \left(1 + \frac{\left((G_p + \xi'_p) - \mu_k \right) (G_q - \mu_k)}{\sigma_k^2 + \varepsilon} \right) \tag{11}$$

where ξ'_p is the added offset and ε is the smoothing parameter. The näive offset choosing strategy is also applied to our proposed AGF, as does ABF. That is:

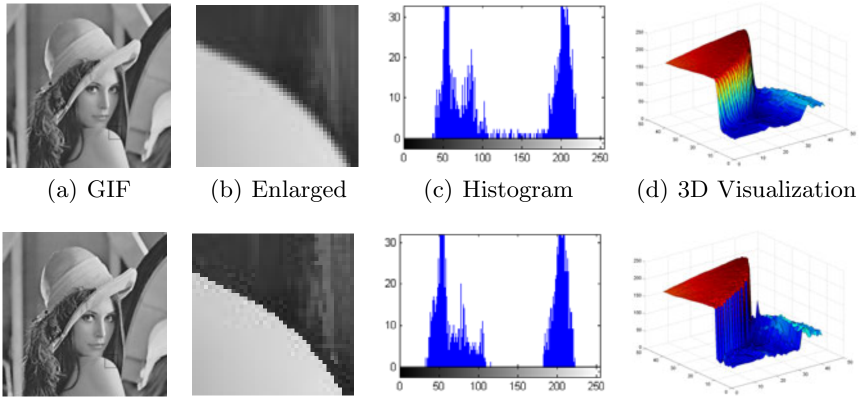
$$\xi'_p = \begin{cases} \mathrm{MAX}(w_k) - G_p & \text{if } \Delta'_p > 0 \\ \mathrm{MIN}(w_k) - G_p & \text{if } \Delta'_p < 0 \\ 0 & \text{if } \Delta'_p = 0 \end{cases} \tag{12}$$

where the intensity difference is defined by $\Delta'_p = G_p - \mu_k$.

The same histogram analysis is applied to GIF and our proposed AGF, as shown in Fig. 2. GIF only preserves the edges during the smoothing process, while the sharpened result produced from our proposed AGF with the näive offset choosing strategy contains the aliasing effect and unexpected outliers, as did näive ABF. In order to achieve the better result by applying the adaptive optimal parameters produced from [19], the values of ε in AGF need to be computed based on the corresponding optimal values of σ_r in ABF. In Section 2, we showed these two parameters can be converted by the following expression:

$$\varepsilon = \sigma_r^2 / 255 \tag{13}$$

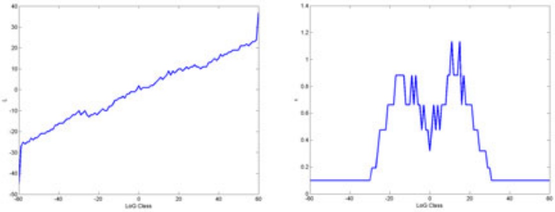
where both ε and σ_r are in the range $[0; 255]$ intensity value. Fig. 3 shows the corresponding offset and converted epsilon values we will use in AGF. The offset tends to be unchanged. However, to make sure the term $G_p + \xi'_p$ is still within the range $[\mathrm{MIN}(w_k); \mathrm{MAX}(w_k)]$, it is constrained by the following equation:



(a) GIF (b) Enlarged (c) Histogram (d) 3D Visualization

(e) näive AGF (f) Enlarged (g) Histogram (h) 3D Visualization

Fig. 2. Illustration of the effect of our proposed AGF ($\varepsilon = 1.5686$) with the näive offset choosing strategy compared to conventional GIF ($\varepsilon = 1.5686$)



(a) Offset ξ (b) Epsilon ε

Fig. 3. Optimal offset and converted epsilon correspoding to each LoG class

$$\xi'_p = \begin{cases} \mathrm{MAX}(w_k) - G_p & \text{if } \mathrm{A}_p > \mathrm{MAX}(w_k) \\ \mathrm{MIN}(w_k) - G_p & \text{if } \mathrm{A}_p < \mathrm{MIN}(w_k) \\ \xi_p & \text{otherwise} \end{cases} \tag{14}$$

where ξ_p is the optimal offset obtained from [19] and $\mathrm{A}_p = G_p + \xi_p$. It's noted that, each pixel p is classified by the corresponding LoG class number obtained by applying a LoG filter. The rounded LoG class is limited within the range $[-60; 60]$ as does ABF.

4.2 Linear Transform Model of AGF

In this section, we present AGF in terms of the linear translation-variant, because the $O(N)$ time exact algorithm takes advantage of this model to implement it. First, we will show the linear transform model of GIF, and then apply it to the proposed AGF. As described in [9], the filtered output $\hat{I}$ of GIF is represented by a linear transform of guided image G within a local window w_k centered at pixel k as follows:

$$\hat{I}_p = a_k G_p + b_k, \forall p \in w_k \tag{15}$$

where a_k and b_k are constant linear coefficients determined by solving the optimization problem that seeks to minimize the difference between the output and input image. Formally, it is expressed by:

$$E(a_k, b) = \sum_{p \in w_k} \left((a_k G_p + b_k - I_p)^2 + \varepsilon_k a_k^2 \right) \tag{16}$$

where ε_k is unchanged over the entire image. It controls the degree of smoothing of GIF. These coefficients are formally determined using linear regression method:

$$a_k = \frac{\frac{1}{|w|} \sum_{p \in w_k} G_p I_p - \mu_k \bar{I}_k}{\sigma_k^2 + \varepsilon_k} \tag{17}$$

$$b_k = \bar{I}_k - a_k \mu_k \tag{18}$$

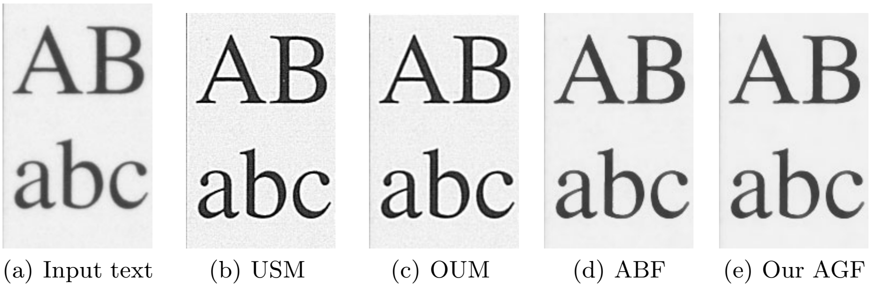
where $\bar{I}_k$ is the mean of I in w_k. To ensure the value of $\hat{I}_p$ does not vary when computed in different windows, the final output is computed by:

$$\hat{I}_p = \left(\frac{1}{|\mathrm{w}|} \sum_{k \in w_p} a_k \right) G_p + \left(\frac{1}{|w|} \sum_{k \in w_p} b_k \right) \tag{19}$$

For our proposed AGF, the question is how to include the adaptive optimal parameters into the linear transform-variant of the GIF. First, we can clearly see the varying adaptive ε^* obviously fits well to equation (17) when computing linear coefficient a_k. Second, the function

$$\hat{I}_p = \left(\frac{1}{|\mathrm{w}|} \sum_{k \in w_p} a_k \right) (G_p + \xi'_p) + \left(\frac{1}{|w|} \sum_{k \in w_p} b_k \right) \tag{20}$$

is the linear transform model of AGF with the participation of the adaptive offset. The appendix presented at the end of this paper shows the correspondence between this linear transform model and its filter kernel expressed in equation (10) and (11). Hence, the algorithm can be implemented by applying a chain of box filters using $O(N)$ integral image technique, as does GIF.

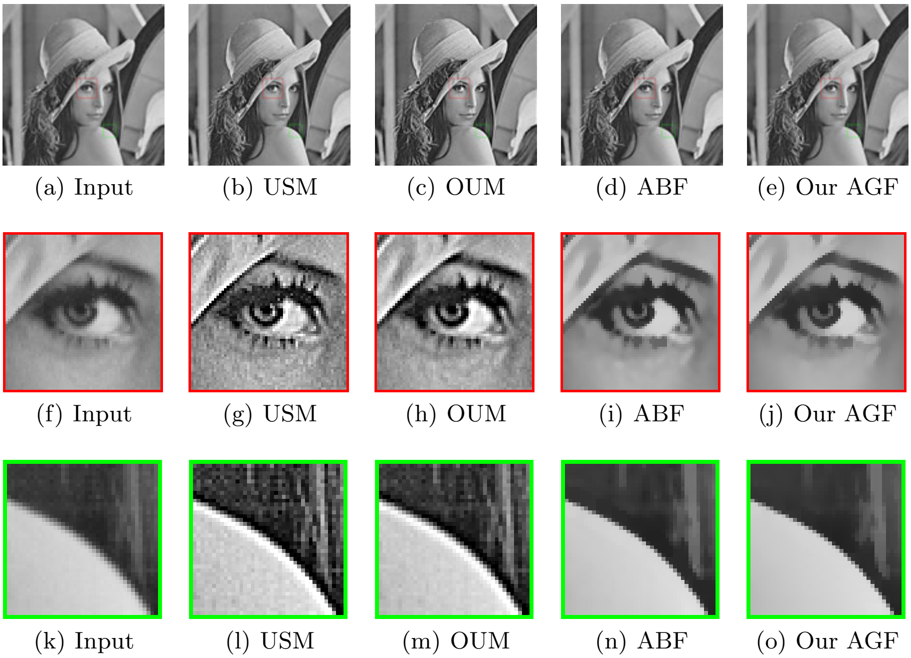


(a) Input text (b) USM (c) OUM (d) ABF (e) Our AGF

Fig. 4. Scanned text image rendered by our proposed AGF and existing methods. Parameters are configured as follows: (b) USM: $r = 5, \lambda = 4$; (c) OUM: $r = 5$; (d) ABF: $r = 3, \sigma_s = 1.0, r_{LoG} = 4$; (e) AGF: $r = 3, r_{LoG} = 4$.

5 Experimental Results

We evaluate the performance of AGF and existing methods with a scanned text image and the Lena image. The text image scanned at 600 dpi was obtained from [19] and cropped due to space limitations. For the text image, as shown in Fig. 4, the contrast of restored outputs produced from USM and OUM increase; but visible halos occur around the edges. Conversely, restored texts produced from ABF and our AGF do not suffer such artifacts, and the contrast is nearly identical to that of the input image. For the Lena image, the difference between these methods can be seen more clearly, as shown in Fig. 5. USM produces visible halos around the edges, and the noise is also significantly enhanced. OUM reduces noise but suffers from the artifacts. Both ABF and our AGF with the use of optimal adaptive parameters effectively remove noise and significantly enhance the sharpness. We used a PC with an AMD Athlon 64 X2 Dual Core Processor 3800+ 2.00 Ghz to measure the processing time of both AGF and ABF with a kernel radius $r = 5$. Our proposed AGF takes about 1.4s to process a 1-megapixel gray-scale image, while the $O(|w|^2)$ time ABF [19] takes about 12.7s to process it.



(a) Input (b) USM (c) OUM (d) ABF (e) Our AGF

(f) Input (g) USM (h) OUM (i) ABF (j) Our AGF

(k) Input (l) USM (m) OUM (n) ABF (o) Our AGF

Fig. 5. Lena image rendered by our proposed AGF and existing methods. Parameters are configured as follows: (b), (g), (l) USM: $r = 5, \lambda = 4$; (c), (h), (m) OUM: $r = 5$; (d), (i), (n) ABF: $r = 3, \sigma_s = 1.0, r_{LoG} = 4$; (e), (j), (o) AGF: $r = 3, r_{LoG} = 4$.

6 Conclusion

In this paper, we presented an adaptive guided image filtering (AGF) for sharpness enhancement and noise reduction. The proposed method is developed based on guided image filtering and the shift-variant technique. The relationship between the conventional bilateral filter and the guided image filter is presented to convert optimal parameters from ABF to our proposed AGF.

Experiments showed the results produced from our method to be superior to those produced from unsharp masking-based techniques and comparable to ABF filtered output. It effectively removes noise and sharpens the edges simultaneously, without producing overshoot and undershoot artifacts as the ideal approach. Our method outperforms ABF in terms of computation cost, where the computational complexity is $O(N)$ compared to $O(|w|^2)$ of ABF.

References

1. Aurich, V., Weule, J.: Non-linear gaussian filters performing edge preserving diffusion. In: Proceedings of the DAGM Symposium, pp. 538–545 (1995)
2. Barash, D.: A Fundamental Relationship Between Bilateral Filtering, Adaptive Smoothing, and the Nonlinear Diffusion Equation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(6), 844–847 (2002)
3. Bilcu, R.C., Vehvilainen, M.: Constrained Unsharp Masking for Image Enhancement. In: Proc. of Intl. Conf. on Image and Signal Processing, pp. 10–19 (2008)
4. Buades, A., Coll, B., Morel, J.M.: The staircasing effect in neighborhood filters and its solution. IEEE Trans. Image Processing 15(6), 1499–1505 (2006)
5. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. ACM Transactions on Graphics 26(3) (2007)
6. Crow, F.C.: Summed-area tables for texture mapping. In: SIGGRAPH (1984)
7. Durand, F., Dorsey, J.: Fast Bilateral Filtering for the Display of High-Dynamic-Range Images. ACM Transactions on Graphics 21(3), 257–266 (2002)
8. Elad, M.: On the bilateral filter and ways to improve it. IEEE Transactions on Image Processing 11(10), 1141–1151 (2002)
9. He, K., Sun, J., Tang, X.: Guided Image Filtering. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 1–14. Springer, Heidelberg (2010)
10. Kim, S., Allebach, J.P.: Optimal unsharp mask for image sharpening and noise removal. Journal of Electronic Imaging 14, 023007-1–023007-13 (2005)
11. Paris, S., Durand, F.: A Fast Approximation of the Bilateral Filter using a Signal Processing Approach. International Journal of Computer Vision 81(1), 24–52 (2009)
12. Paris, S., Kornprobst, P., Tumblin, J., Durand, F.: Bilateral Filtering: Theory and Applications. In: Foundations and Trends in Computer Graphics and Vision (2009)
13. Pham, T.Q., Van Vliet, L.J.: Separable bilateral filtering for fast video preprocessing. In: Proceedings of the IEEE Intl. Conf. on Multimedia and Expo (2005)
14. Penora, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)

15. Polesel, A., Ramponi, G., Mathews, V.G.: Image Enhancement via Addaptive Unsharp Masking. IEEE Trans. Image Processing 9(3), 505–510 (2000)
16. Smith, S.M., Brady, J.M.: SUSAN - A new approach to low level image processing. International Journal of Computer Vision 23(1), 45–78 (1997)
17. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the IEEE Intl. Conf. on Computer Vision (ICCV), pp. 839–846 (1998)
18. Zhang, B., Allebach, J.P.: Adaptive Bilateral Filter for Sharpness Enhancement and Noise Removal. In: Proc. Intl. Conf. on Image Processing (ICIP), vol. 4, pp. 417–420 (2007)
19. Zhang, B., Allebach, J.P.: Adaptive Bilateral Filter for Sharpness Enhancement and Noise Removal. IEEE Transactions on Image Processing 17(5), 664–678 (2008)

Appendix: Derivative of the AGF Filter Kernel

This is based on the proof that shows the filter kernel of GIF corresponds to its linear translation-variant in [9], we shortly present the correspondence between the filter kernel and linear transform model of AGF with the introduction of optimal offset ξ^* and ε^* in this part.

First, we rewrite equation (10) by $\hat{I}_p = \sum_{q \in w_k} W_{AGF_{pq}}(G) I_q$. So, the filter kernel $W_{AGF_{pq}}(G)$ is computed by taking the partial derivative of $\hat{I}_p$ with respect to I_q. Formally, it is expressed by:

$$W_{AGF_{pq}}(G) = \frac{\partial \hat{I}_p}{\partial I_q} \tag{21}$$

Replacing b_k in (20) by (18), we have:

$$\hat{I}_p = \frac{1}{|w|} \sum_{k \in w_p} \left[a_k \left((G_p + \xi'_p) - \mu_k \right) + \bar{I}_k \right] \tag{22}$$

So, the partial derivative of $\hat{I}_p$ with respect to I_q is formulated by:

$$\frac{\partial \hat{I}_p}{\partial I_q} = \frac{1}{|w|} \sum_{k \in w_p} \left[\frac{\partial a_k}{\partial I_q} \left((G_p + \xi'_p) - \mu_k \right) + \frac{\partial \bar{I}_k}{\partial I_q} \right] \tag{23}$$

From [9], we already had:

$$\frac{\partial a_k}{\partial I_q} = \frac{1}{\sigma_k^2 + \varepsilon_k} \left(\frac{1}{|w|} G_q - \frac{1}{|w|} \mu_k \right) \delta_{k \in w_q} \tag{24}$$

$$\frac{\partial \bar{I}_k}{\partial I_q} = \frac{1}{|w|} \delta_{q \in w_k} = \frac{1}{|w|} \delta_{k \in w_q} \tag{25}$$

where $\delta_{q \in w_k}$ equals 1 when q is in w_k, and equals 0 otherwise.

Placing (24) and (25) into (23), we get:

$$\frac{\partial \hat{I}_p}{\partial I_q} = \frac{1}{|w|^2} \sum_{k:(p,q) \in w_k} \left(1 + \frac{\left((G_p + \xi'_p) - \mu_k \right) (G_q - \mu_k)}{\sigma_k^2 + \varepsilon_k} \right) \tag{26}$$

This is exactly the filter kernel $W_{AGF_{pq}}(G)$ that we expressed in equation (11).

Half-Sweep Imaging for Depth from Defocus

Shuhei Matsui, Hajime Nagahara, and Rin-ichiro Taniguchi

Graduate School of Information Science and Electrical Engineering, Kyushu University.
744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan
{matsui,nagahara,rin}@limu.ait.kyushu-u.ac.jp

Abstract. Depth from defocus (DFD) is a technique to recover the scene depth from defocusing in images. DFD usually involves two differently focused images (near-focused and far-focused) and calculates the size of the depth blur in the captured images. In recent years, the coded aperture technique, which uses a special pattern for the aperture to engineer the point spread function (PSF), has been used to improve the accuracy of DFD estimation. However, coded aperture sacrifices an incident light and loses a SNR of captured images which is needed for the accurate estimation. In this paper, we propose a new computational imaging, called half-sweep imaging. Half-sweep imaging engineers PSFs for improving DFD and maintaining the SNR of captured images. We confirmed the advantage of the imaging in comparison with conventional DFD and coded aperture in experiments.

Keywords: computational photography, depth from defocus, image deblurring.

1 Introduction

There are many methods, referred to as depth from defocus (DFD) techniques [9], [11], for estimating scene depths using a single camera. The methods use depth blurs (i.e., blurring that depends on the scene depth) that appear in captured images. DFD usually employs a pair of images one being near-focused and the other being far-focused to determine differences in sizes of depth blurs resulting from depth differences in a scene. However, the circular shape of the aperture of a regular camera is not beneficial for DFD estimation, since the aperture moderately affects depth blurring. For more robust DFD estimation, many researchers have investigated coded aperture techniques [3], [5], [13]. Such techniques use special patterns for the camera aperture to control the shape of the point spread function (PSF). Additionally, it is well known that the shape of the PSF directly affects the frequency response of an imaging system, which is described by the optical transfer function in the field of optics. We can select aperture patterns that drastically change the PSF shape in the image domain or its frequency response in the Fourier domain according to scale changes of the PSF due to object depth differences, thus achieving more accurate DFD estimation in discriminating scene depths. However, the use of a coded aperture attenuates the intensity of captured images, since incident light from the scene is blocked in engineering the PSFs. The attenuation decreases the signal-to-noise ratio (SNR) of the images and limits the improvement of DFD estimation.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 335–347, 2011.

In this paper, we propose a new imaging operation called half-sweep imaging for DFD estimation. DFD has sometimes ignored the quality of the recovering image. We focus to realize high quality of all-in-focus image reconstruction as well as robust DFD estimation for considering to visualization in computational photography. The technique is inspired by focus sweeping [7], [4] and is extended to DFD applications. Half-sweep imaging obtains two images by sweeping the focus during the image exposure time. It has the advantage of a higher SNR for captured images, since we can engineer the image PSFs even if a camera aperture is open. The operation requires the continuous changing of the lens focus or sweeping of an image sensor, which is easy to implement since we can utilize an auto-focusing mechanism or an actuator for image stabilization that current commercial cameras already possess. Moreover, the method has complete compatibility with regular imaging and adaptivity to scene depth when we stop the sweeping motion or freely adjust the sweeping length and positions. Employing the proposed method, we integrate multiple PSFs with different focus settings obtained by focal sweeping to control the frequency responses of imaging PSFs. We split a sweep into half regions to capture images. The two obtained images are captured for the same scene, but using different PSFs (i.e., transfer functions of imaging). As a result, one of the PSFs and captured images has zero-crossing in its frequency response, which helps with depth estimation, and the sum of PSFs has a broadband spectrum, which allows recovery of a better all-in-focus image.

2 Related Work

Many researchers have proposed PSF engineering methods to improve DFD estimation. As an early work on coded apertures, Hiura and Matsuyama [3] used three or four pin holes as the aperture of a multiple-focus camera. They used three differently focused images captured by the camera and realized robust depth estimation. However, this aperture coding was far from optimal.

Levin et al. [5] proposed using an aperture with a pattern more distinguishable than that of a conventional circular aperture. They defined K-L divergence as a metric of the PSF scale difference due to depth difference and found an optimal pattern for DFD estimation by maximizing the metric. The Fourier spectrum of the pattern contains many zero-crossings and their positions are displaced when the blur size changes owing to the depth difference. If we use a different size of the PSF for deconvolution, the recovered image has severe artifacts from the disagreement with the true PSF spectrum. The artifacts increase the penalty for misrecognizing the depth and improve the stability of DFD estimation. As a result, they allow DFD estimation from a single image, while common DFD methods require at least two differently focused images to solve ambiguity in the blurred image due to texture. However, the aperture is not suited to recovering an all-in-focus image through deconvolution, since the frequency response of a zero-crossing point is such that we have zero information at that frequency.

Zhou et al. [13] proposed a coded aperture pair to recover a high-quality focused image and estimate depth. It is well known that a broadband PSF in the Fourier domain is favorable for blurred-image recovery through deconvolution, since it provides image information through the entire frequency range even though the captured image is blurred [12], [14]. However, as mentioned for Levin et al.'s work [5], zero-crossings are favorable for depth estimation. These properties are not compatible with each other when only using a single aperture pattern. There is a dilemma in practical DFD applications that it is necessary to recover the true texture for accurate depth estimation, but recovering the texture requires knowledge of the correct depth information. Therefore, Zhou et al. [13] proposed the use of a pair of coded apertures that optimize image reconstruction and depth estimation simultaneously. In the case of their proposed aperture pair, the frequency response of a single PSF has zero-crossings, but the sum of PSFs has a broadband since the PSFs have complementary responses. The need to replace two lenses with the coded aperture pair remains a difficult problem in image capturing.

A programmable-aperture camera that can quickly switch aperture patterns has been developed [8]. Green et al. [2] proposed a multiple-aperture camera that uses special mode mirrors. There are examples of implementations that have realized easy capturing and increasing flexibility for multiple coded apertures. However, PSF engineering using a coded aperture has an intuitive problem that the SNR of the image is lower than that of the conventional DFD measurement, since the aperture blocks incident light in controlling the PSF shape. Therefore, there is the limitation that noise in the image destabilizes depth estimation and contaminates the recovered image.

Wavefront coding engineers the PSF without blocking incident light unlike the case for a coded aperture. Employing this method, a special optical element called a phase plate is placed at the position of the camera aperture. The phase plate controls the wavefront of rays according to the positions in aperture open. Dowski et al. [1] proposed a phase plate for DFD estimation whose PSF spectrum has many zero-crossings. Levin et al. [6] theoretically analyzed the upper bound of the PSF response for image deblurring and designed optics called a lattice focus lens to realize the PSF. The lens can be used to estimate the scene depth and achieve optimal defocus deblurring, since the PSF of the lattice focus lens is depth-variant. Wavefront coding engineers the PSF with an open aperture and realizes image acquisition with a higher SNR. However, the cost of the phase plate is expensive and its property is not adaptive to a scene.

Nagahara et al. [7], [4] proposed focus sweep imaging that moves focus points during the image integration time to capture a single image. This method integrates different scales of PSFs to realize PSFs that have broadband frequency response and invariant shapes through the entire scene depth. They proposed applying this imaging operation to an extended depth of field by deblurring without any depth estimation or knowledge. The advantages of the focal sweep are a higher SNR of captured image, compatibility of regular photograph and flexibility for scene. Hasinoff et al. [10] discussed the optimal number of focal stack images across a scene depth for various imaging systems. They applied focal sweep imaging to acquire focal stack (multiple) images to obtain a best all-in-focused image. They showed it in simulation and did not compaired DFD accuracy in the paper.

3 Half-Sweep Imaging

Focus sweep imaging [7], [4] sweeps the focal plane through a scene during the image exposure time. It is achieved by moving the lens or image sensor position along the optical axis. We can manipulate the PSF by controlling the range or speed of the

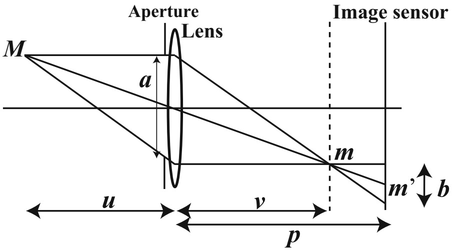


Fig. 1. Projective geometry of lens

sweeping. In this paper, we propose an extension of focal sweep imaging called half-sweep imaging for DFD application. Full-sweep imaging [7], [4] sweeps the focal plane through the entire depth of the target scene in the exposure time to realize the extended depth of field. Our half-sweep imaging splits the sweep range into two regions and captures two images corresponding to the front half and back half of the sweeping regions. Consequently, we capture two images that have depth-variant blurs (PSFs) for DFD estimation, while the original full sweep obtains depth-invariant blurs for deblurring. In this section, we present the properties and advantages of PSFs in half-sweep imaging.

Figure 1 shows the projective geometry where the image sensor is at a distance p from a lens with focal length f, and the aperture diameter is a. Incident rays from a scene point M at the distance u converge to the focused point m at a distance v from the lens. The relation between u and v is described by the Gaussian lens law:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}. \tag{1}$$

As shown in the figure, if an image sensor is placed at a distance p from the lens, M is imaged to m' with blur on the sensor. The diameter of the blurred circle b is given by

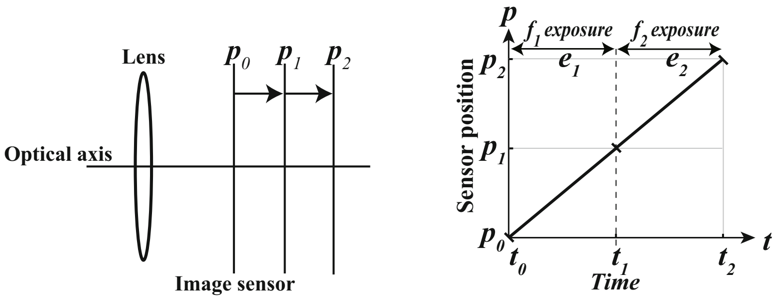
$$b(p) = \frac{a}{v}|(v-p)|. \tag{2}$$

The PSF is a function of the distribution of light energy within the blurred circle. We consider here r to be the distance of an image point from the center m' of the blurred circle, and the PSF is denoted $P(r,u,p)$. The PSF is often modeled as a pillbox function:

$$P(r,u,p) = \frac{4}{\pi b^2}\prod(\frac{r}{b}), \tag{3}$$

where $\prod(x)$ is the rectangle function, which has a value 1 if $|x| < 1/2$ and 0 otherwise. This is the PSF function of an object placed at u when the sensor position is fixed at p as in regular imaging with a common camera.

In half-sweep imaging, the sensor moves from p_0 to p_2 along the optical axis of the camera as shown in Figure 2-a. We assume that focus points of all objects in a scene lie between p_0 and p_2. The half-sweep imaging captures two images f_1 and f_2 with exposures e_1 and e_2 as shown in Figure 2-b. The sensor motion is modeled as a function



a. Sensor motion

b. Motion and image integrations

Fig. 2. Half sweep imaging

of time $p(t) = st + p_0$ if the sensor moves with constant speed s. The relation between the sensor motion and exposure time is shown in Figure 2-b. This figure describes that the exposures e_1 and e_2 for capturing the images f_1 and f_2 correspond to sweep regions from p_0 to p_1 and from p_1 to p_2 respectively. Hence, we obtain two images with different integrations of different blurred images focusing at positions between p_0 and p_1 or p_1 and p_2. It is easy to realize half sweeping by simply changing the shutter timings and the exposure time from those for full sweeping. An imaging process can be modeled by convolution of the PSF function:

$$f_i = h_i \otimes f_0 + \xi, \quad i = 1, 2, \tag{4}$$

where f_i is the observed image, h_i is the half-sweep PSF, f_0 is the latent in-focus image and ξ is the image noise, which is assumed to be Gaussian white noise $N(0, \sigma^2)$. Normally, the shape of a PSF is determined by the aperture size and object depth as described by Equation 3. Meanwhile, the half-sweep PSF h_i is modeled by integration of PSFs at multiple sensor positions p through the sweeping regions during the exposure time. This is described by

$$h_i(r, u) = \int_{p_{i-1}}^{p_i} P(r, u, p) dp, \quad i = 1, 2, \tag{5}$$

where p_i $(i = 0, 1, 2)$ is the position of the image sensor. The sensor moves from p_{i-1} to p_i during exposure time e_i. If we assume that the integrated blur model is a pillbox function as described in Equation 3, the half-sweep PSF is modeled by

$$h_i(r,u) = \frac{uf}{(u-f)\pi a s p_i} \left(\frac{\lambda_{p_{i-1}} + \lambda_{p_i}}{r} - \frac{2\lambda_{p_{i-1}}}{b(p_{i-1})} - \frac{2\lambda_{p_i}}{b(p_i)} \right), \; i = 1, 2, \tag{6}$$

where $b(p)$ is the diameter of the blurred circle at position p, and λ_p = 1 if $b(p) \geq 2r$ and 0 otherwise.

Figure 3-a shows simulated half-sweep PSFs h_1, h_2 modeled by Equation 6 and their average PSF h_{all} at four different scene depths. The four different depth positions u are decided by the relation of the lens law so that the corresponding focal position v is at constant intervals on the sensor side. The average PSF h_{all} is simply derived according

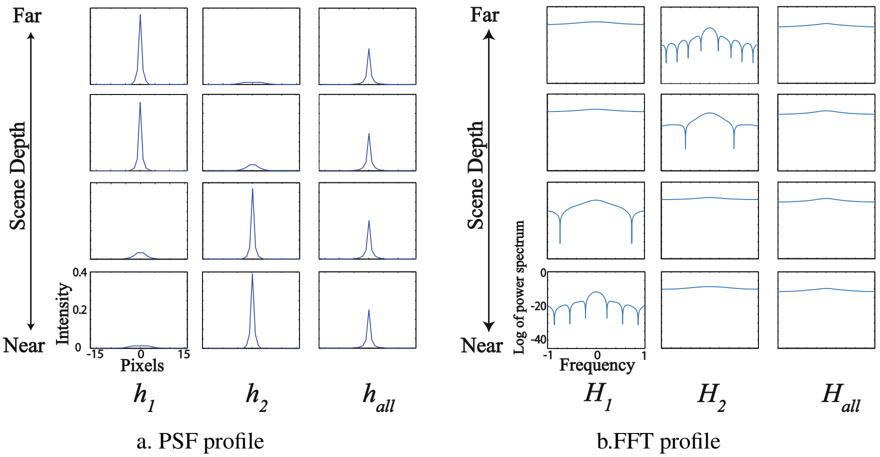


Fig. 3. Half sweep PSF

to $h_{all} = (h_1 + h_2)/2$. Therefore, we consider h_{all} to be the same as the full sweep PSF [7], [4], since the sum of h_1 and h_2 is an integration of different focal images through the entire swept region. The figure shows that PSF shapes h_1 and h_2 change according to the depth difference, while shape of the average PSF h_{all} does not vary visually. Figure 3-b shows the logarithms of power spectrums of these three PSFs with different scene depths in the frequency domain, where H_1, H_2 and H_{all} are the discrete Fourier transforms of h_1, h_2 and h_{all} respectively. This plot corresponds to Figure 3-a. The spectrums of H_1 and H_2 change according to depth. We also see zero-crossings in one of the spectrums. On the other hand, H_{all} has a broadband spectrum. Levin et al. [5] and Zhou et al. [13] claimed that PSFs having zero-crossings are a useful property of the DFD measurement and improve depth discrimination. Additionally, it is well known that broadband PSFs are beneficial for defocus deblurring [12], [14], [7], [4], [13] and allow the generation of good quality all-in-focus images in DFD application.

4 DFD Algorithm for Half-Sweep Imaging

In this section, we propose a method for estimating a depth map and an all-in-focus image from two images captured by half-sweep imaging. Half-sweep imaging is expressed as Equation 4. This can be written in the Fourier domain as

$$F_i^{(d)} = F_0 \cdot H_i^{(d)} + N, \quad i = 1, 2, \tag{7}$$

where $F_i^{(d)}$ is the Fourier transform of a captured images ($i = 1, 2$) at depth d, F_0 is the Fourier transform of a latent all-in-focus image, $H_i^{(d)}$ ($i = 1, 2$) is the Fourier transform of a half-sweep PSFs at depth d and N is the Fourier transform of noise. We consider here the problem in which we estimate the all-in-focus image F_0 and unknown scene depth d from Equation 7. Generally, the image F_0 is given by deconvolution. We use the Wiener deconvolution:

$$\hat{F_0} = \frac{F \cdot \overline{H}}{|H^2| + |C|^2}, \tag{8}$$

where $\bar{H}$ is a complex conjugate of H and $|H|^2 = H \cdot \bar{H}$. C represents $\sigma / A^{\frac{1}{2}}$, where A is defined over the power distribution of natural images according to the $1/f$ law. The original Wiener deconvolution was designed to deblur one blurred image, and we propose to use the extended method in our half-sweep imaging. As shown in section 3, h_{all}, which is the average of h_1 and h_2 kernels, has a broadband frequency response for each depth. Additionally, f_{all}, which is the average image of f_1 and f_2, has broadband image information, since we can assume that the image f_{all} is captured by the h_{all} kernel. The property of addition is maintained over the Fourier transform. Hence, we obtain the Fourier transforms of the average kernel and the image as

$$F_{all} = \frac{F_1 + F_2}{2}, \quad \overline{H_{all}^{(d)}} = \frac{\overline{H_1^{(d)}} + \overline{H_2^{(d)}}}{2}. \tag{9}$$

We can extend Wiener deconvolution to half-sweep imaging by substituting Equation 9 into Equation 8:

$$\hat{F_0}^{(d)} = \frac{(F_1 + F_2)\overline{(H_1^{(d)} + H_2^{(d)})}}{|H_1^{(d)} + H_2^{(d)}|^2 + 4|C|^2}. \tag{10}$$

We consider that the error between observed images and estimated observed images must be minimum when the estimated depth d is correct. Therefore, we defined a cost function to estimate depth d is expressed as

$$W^{(d)} = \sum_{i=1,2} |IFFT(\hat{F_0}^{(d)} \cdot H_i^{(d)} - F_i)|, \tag{11}$$

where $IFFT$ is the 2D inverse Fourier transform and $\hat{F_0}$ is derived from Equation 10. The cost function $W^{(d)}$ represents the error between the reconstructed images and the captured images; therefore, $W^{(d)}$ is a measure of how close d is to the actual scene depth d^*. We estimate depths to find the minimum $W^{(d)}$ for each pixel (x, y) using

$$U(x, y) = \arg\min_{d \in D} W^{(d)}(x, y). \tag{12}$$

We also obtain an all-in-focus image I from the estimated depth map U as

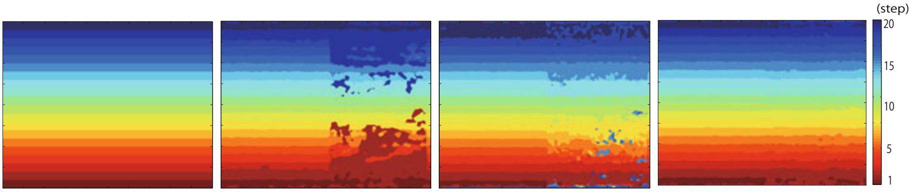
$$I(x, y) = \hat{F_0}^{(U(x,y))}(x, y). \tag{13}$$

5 Performances Analysis

We carried out simulation experiments to evaluate the performance of our half-sweep imaging. In this section, we denote the object position as u, the focal point v and sensor position p as shown in Figure 1. We assumed that the scene is a synthetic staircase

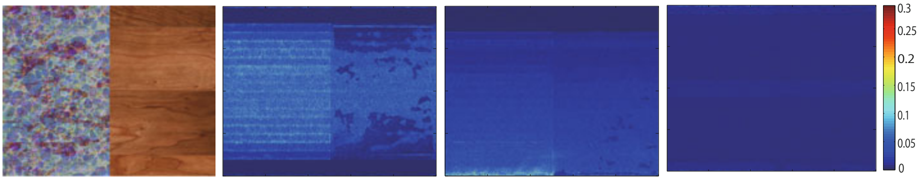
Table 1. Correspondence among step number, object depth and focus position (f = 9 mm)

Depth step	Step1	Step5	Step10	Step15	Step20	Step25	Step30	Step35	Step40
Object:$u_{\text{[mm]}}$	2034	321	160	109	83.6	68.6	58.5	51.4	46.1
Focus: $v_{\text{[mm]}}$	9.040	9.260	9.535	9.810	10.085	10.360	10.64	10.91	11.185



a. Ground Truth b. Conventional DFD c. Coded aperture pair d. Half-sweep imaging

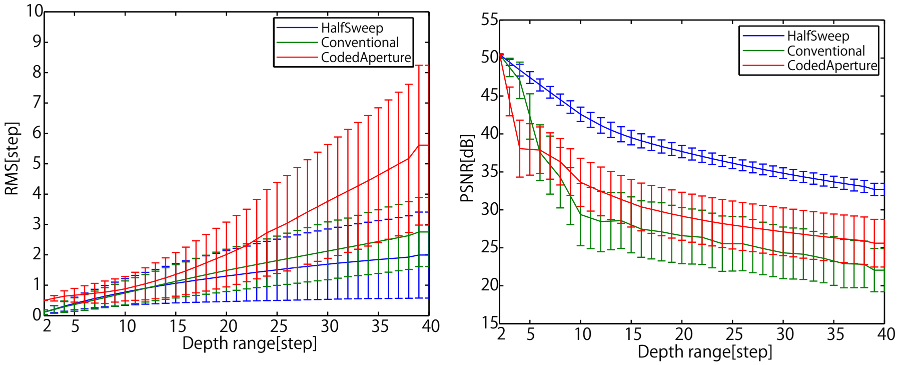
Fig. 4. Estimated depth map



a. True texture b. Conventional DFD c. Coded aperture pair d. Half-sweep imaging

Fig. 5. Error map of deblurred image

scene as shown in Figure 4-a. These depth maps have false-color representation with red indicating locations far (step 1) from the camera and blue locations near (step 20) the camera. The scene has two textures, one with strong and dense patterns and the other of natural wood with weak texture as shown in Figure 5-a. The physical object depths u are from 2034 to 83.6 mm from a camera lens. The corresponding focal point v varies from 9.04 to 10.085 mm behind the lens according to the lens law of Equation 1. We divided the possible range of the focal position v into 20 uniform steps (Δv =0.055 mm) so that the depth blur must change by a similar ratio (0.5 pixels for each step) in an image. Table 1 gives the conversion among a step number, the corresponded object depth u and the focal position v for easy understanding of the relations. The focal length and F-number of a lens are taken as f = 9 mm and $f/1.4$ in our setting. Under these settings, we simulated captured images through convolution with theoretical PSFs modeled by the pillbox function described by Equation 6. We set integration intervals to half by half of the target depth range for half-sweep DFD. When the depth range is 20 steps in this case, the intervals are 1 to 10 and 11 to 20 steps. The corresponding sensor positions are p_0=9.04 mm, p_1=9.945 mm and p_2=10.085 mm for equation 5. The conventional DFD used far-focused (step 1, p = 9.04 mm) and near-focused (step 20, p = 10.085 mm) images with an open circular aperture. The coded aperture used two far-focused (step 20, p = 9.04 mm) images captured by an aperture pair [13](i.e., aperture difference is the depth key). We estimated scene depth maps and all-in-focus images using the proposed DFD algorithm as mentioned in Section 4. For the conventional and coded-aperture DFD estimation, we used Zhou's DFD algorithm [13] for comparison.



a. RMS vs Depth range

b.PSNR vs Depth range

Fig. 6. Simulation results

Figure 4 shows the depth estimation results. The figure shows the depth maps of (a) the ground truth, (b) the conventional DFD method, (c) Zhou's coded aperture pair and (d) our proposed method by capturing half-sweep imaging. We see that the strong texture on the left side of the scene does not differ greatly among the methods. However, there are large differences for the weak texture on the right side, with our proposed half-sweep imaging having the best performance. Figure 4-b, the result for conventional DFD estimation, shows large error around the central depth, and Figure 4-c, the result for coded-aperture-pair DFD estimation, shows error for the entire depth range. On the other hand, Figure 4-d, the result for our method, shows greater robustness, although the scene has weak texture. Figures 5-b, c, d show difference images between the estimated images and the true texture as shown in Figure 5-a, since it is difficult to recognize the error in the estimated images. The figures are shown by false color representation and the color bar indicates the errors in normalized intensity (i.e., maximum intensity is 1.0). Figure 5-b shows large reconstruction errors for the center of the image, since captured images have large blurs and high-frequency information was lost in the center of the image in conventional DFD estimation. Figure 5-c shows that the recovery errors increase where the object depth approaches a far position. It is difficult to distinguish the difference between the coded aperture pair where the size of blur is small or in focus, since the method of the coded aperture pair employs the shape difference between the apertures as a depth key. Figure 5-d shows that the proposed recovery method produces errors that are smaller and more uniform.

We also compared numerical qualities among these methods. In this experiment, we used similar setting to Figure 4, but we changed the object depth range (the number of stairs) from 2 to 40 steps. Table 1 also shows the conversion of the object depth. We used arbitrary 30 images downloaded from $flickr$ as scene textures for generating simulated images. Figure 6-a and b show the root-mean-square (RMS) of the estimated depth errors and the peak signal-to-noise ratios (PSNR) of the recovered images against the object ranges. In these figures, line plots indicate the average values of the RMS

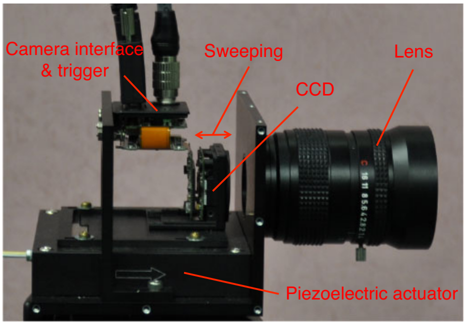


Fig. 7. Prototype focal sweep camera

error or PSNR, and the error bars indicate the standard deviations of 30 variations of the textures. The standard deviations imply that the each result is deviated depending on the texture difference. Figure 6-a shows that all of the methods are getting worse if the depth range are enlarged, since bigger size of blur must be used for estimating larger depth range and it is difficult to estimate the blur size when the size is larger. Half sweep DFD is still better performance for estimating the depth than the others. Figure 6-b also shows the similar results that PSNR is getting worse when the depth range is enlarged among the all methods. Yet, it is obvious that the PSNR of the proposed method is far better than that of the others. We can also see that the standard deviation of the proposed method is smaller than the others. It means that half sweep DFD is more robust to recover the images independent to the scene texture variety. These figures show that the proposed method outperforms both depth estimation and the recovered image quality.

We confirm that the proposed method of half-sweep imaging has the best performance in terms of estimating the scene depth and recovering an all-in-focus image. This is due to one of the proposed half-sweep imaging PSFs having zero-crossings and their sum having a broadband spectrum in the Fourier domain.

6 Real Implementation and Experiments

We evaluated our half-sweep imaging for real images captured by a prototype camera. Figure 7 shows the prototype camera for realizing half-sweep imaging. The camera consists of a 1/3" Sony CCD (with 1032×776 pixels) mounted on a Physik Instrument P-628.1CL translation stage. This stage is driven by a piezoelectric actuator and the range of translation is 800 microns. We attached a Tokina 12.5 mm lens and the F-number was set to $f/1.4$ in this experiment. The shutter of the CCD and the actuator were controlled with by signals generated by PC. They were completely synchronized for realizing half-sweep imaging.

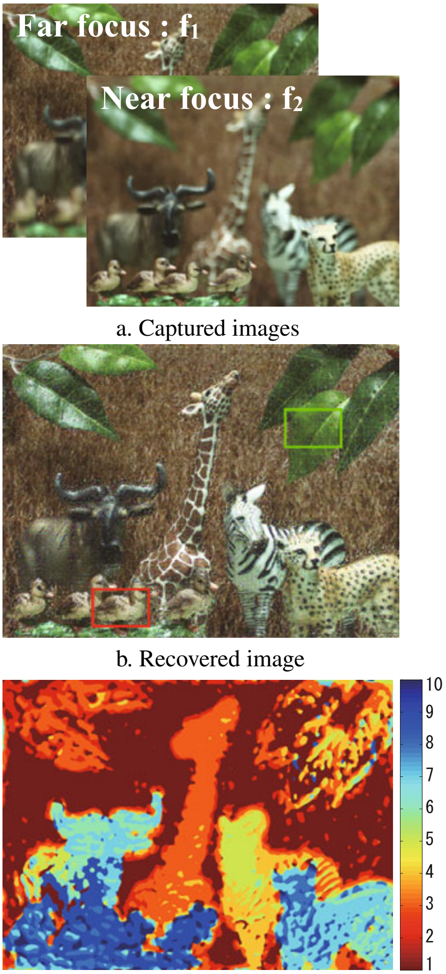
A target scene that we captured in this experiment is shown in Figures 8 and 9. There are four objects at different depths in front of a wall in the scene. The range of scene depth was 340 to 750 mm from the camera lens. The actuator needed to translate 225 microns from start to end positions to covering the entire scene depth. Figure 9-a shows the images captured using the prototype camera with half-sweep imaging. The images f_1 and f_2 were captured by the front half and back half of the sensor sweeping. Figures 8-a show images again captured using the prototype camera but with near-focus and far-focus positions (not sweeping) for conventional DFD. We captured measured PSFs for both half-sweep imaging and conventional DFD estimation at ten depths using a point light source before the experiments.

We estimated the object depths and recovered all-in-focus images using the input images and measured PSFs. Figures 9-b, c show the results of the recovered all-in-focus images and the depth map of the scene. We employed the proposed DFD method as mentioned in Section 4. Figures 8-b, c show the results of conventional DFD estimation for comparison. Comparing Figures 9-d and 8-d, we see that both depth maps show the depth differences among the objects and we cannot see a strong advantage for one method over the other. It was caused by that the scene has relatively strong texture unlike the scene in the simulation. Hence, the difference was not appeared between the methods.

Comparing Figures 9-b and 8-b, we see that the recovered focused images have large differences. The image obtained through conventional DFD estimation has many artifacts such as enhanced noise and ringing artifacts. We also see that some portion of the texture is still blurred even after deconvolution because of depth estimation errors. Figures 10 clearly show the differences for magnified portions of the images. Figure 10-c shows the ground truth textures captured for the same scene with a small aperture setting of $f/16$. The proposed method does not provide an image that is identical to the ground truth but has far better performance than the conventional DFD approach. The experiments confirm that our proposed method has an advantage over the conventional DFD and works in a real implementation.

7 Conclusion

This paper proposes a new computational imaging technique called half-sweep imaging and a processing method for DFD estimation. The sensor sweeps during the exposure time to capture two images. We show that PSFs of the proposed half-sweep imaging simultaneously have zero-crossings and broadband properties in the Fourier domain. We realized robust depth estimation and high-quality image recovery from the contribution of the PSF properties. We confirmed the advantage of our half-sweep imaging over previous methods in simulation and real experiments. We implemented a prototype camera that incorporates a piezoelectric actuator to sweep an image sensor in real experiments. However, the sweeping operation can be more easily implemented for commercial cameras; e.g., utilizing the auto-focusing mechanism. Half-sweep imaging has the advantages of obtaining a higher SNR for images, having flexibility such that it can adapt to the scene depth, and having complete compatible with regular imaging. Hence, it is applicable for a wide range of products such as digital still cameras.

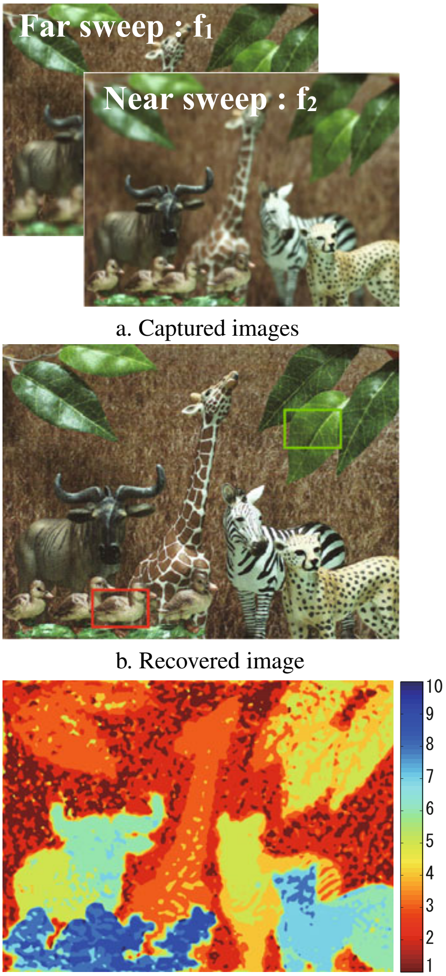


a. Captured images

b. Recovered image

c. Depth map

Fig. 8. Conventional DFD

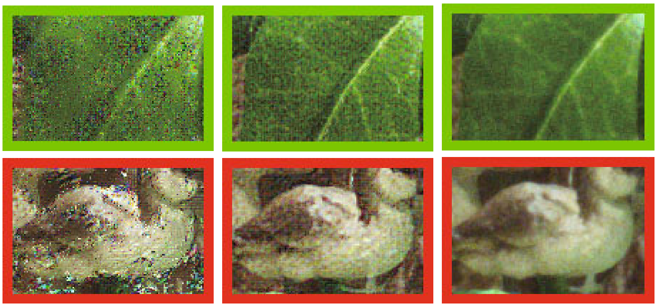


a. Captured images

b. Recovered image

c. Depth map

Fig. 9. Half-sweep imaging



a. Conventional DFD b. Half-sweep imaging DFD c. Ground truth ($f/\#$=16)

Fig. 10. Zoom up potion of images

References

1. Dowski, E.R., Cathey, W.T.: Single-lens single-image incoherent passive-ranging systems. Applied Optics 33(29) (1994)
2. Green, P., Sun, W., Matusik, W., Durand, F.: Multiple-aperture photography. In: Proc. ACM SIGGRAPH (2007)
3. Hiura, S., Matsuyama, T.: Depth measurement by the multi-focus camera. In: Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, pp. 953–959 (1998)
4. Kuthirummal, S., Nagahara, H., Zhou, C., Nayar, S.K.: Flexible depth of field photography. IEEE Trans. Pattern Analysis and Machine Intelligence 33(1), 58–71 (2011)
5. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM Transactions on Graphics 26(3) (2007)
6. Levin, A., Hasinoff, S., Green, P., Durand, F., Freeman, W.T.: 4d frequency analysis of computational cameras for depth of field extension. ACM Transactions on Graphics (2009)
7. Nagahara, H., Kuthirummal, S., Zhou, C., Nayar, S.K.: Flexible Depth of Field Photography. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 60–73. Springer, Heidelberg (2008)
8. Nagahara, H., Zhou, C., Watanabe, T., Ishiguro, H., Nayar, S.K.: Programmable Aperture Camera Using lCoS. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 337–350. Springer, Heidelberg (2010)
9. Pentland, A.: A new sense for depth of field. IEEE Tans. Pattern Analysis and Machine Intelligence 9(4), 423–430 (1978)
10. Hasinoff, S.W., Kutulakos, K.N., Durand, F., Freeman, W.T.: Time-constrained photography. In: Proc. IEEE International Conference on Computer Vision (2009)
11. Subbarao, M., Gurumoorthy, N.: Depth recovery from blurred edges. In: Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, pp. 498–503 (1988)
12. Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., Tumblin, J.: Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. In: ACM SIGGRAPH (2007)
13. Zhou, C., Lin, S., Nayar, S.: Coded aperture pairs for depth from defocus. In: Proc. IEEE International Conference on Computer Vision (2010)
14. Zhou, C., Nayar, S.K.: What are good apertures for defocus deblurring? In: IEEE International Conference on Computational Photography (2009)

A Hierarchical Approach to Practical Beverage Package Recognition

Mei-Chen Yeh and Jason Tai

Department of Computer Science and Information Engineering
National Taiwan Normal University, Taipei, Taiwan
myeh@csie.ntnu.edu.tw, whiteorange0617@gmail.com

Abstract. In this paper we study the beverage package recognition problem for mobile applications. Unlike products such as books and CDs that are primarily packaged in rigid forms, the beverage labels may be attached on various forms including cans and bottles. Therefore, query images captured by users may have a wide range or variations in appearance. Furthermore, similar visual patterns may appear on distinct beverage packages that belong to the same series. To address these challenges, we propose a fast, hierarchical approach that can be used to effectively recognize a beverage package in real-time. A weighting scheme is introduced to enhance the recognition accuracy rate when the query beverage is among flavor varieties in a series. We examine the development of a practical system that can achieve a fairly good recognition performance (93% accuracy rate using an evaluation set of 120 images) in real-time.

Keywords: Product recognition, mobile application, sub-image retrieval.

1 Introduction

A technology that enables customers get product information in an easy, fast, and intuitive manner is essential for cost-conscious shopping [9]. With developments in handsets that have increased the computing and communication capabilities, content-based product recognition is a complementary approach to existing technologies such as Barcodes [14] and Radio Frequency Identification (RFID) [10], in which a tag is required to be attached to each item for identification. A primary advantage of the content-based approaches over tag-based approaches is the fact that recognition can be performed directly from any part of the content—not necessarily barcodes or RFID tags—with a device that may not be equipped with a tag reader.

Among many products of interest, we study the beverage package recognition problem in this paper as the beverage industry has been continuing to be an economic powerhouse [4]. The beverage package recognition problem can be considered a simplified object recognition problem because the patterns of beverage packages are more structured and rigid comparing to those of other objects such as human faces. However, query photos captured by users can still have a wide range of variations in appearance because beverages can be packed in various forms, e.g. boxes, can, and bottles, and a package can be arranged in any angle to users. Figure 1 (a) illustrates three examples of a Coke can. The most recognizable part (e.g. the brand logo) could

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 348–357, 2011.

be fully or partially captured, or totally invisible on the query image. Moreover, it is common in package design that distinct beverage products in the same series share similar visual features. In Fig. 1 (b), these images have identical visual patterns, e.g. brand names printed with the same font. However, they should be considered distinct beverages as they are differently flavored, and the price and calorie information may be different. These factors make the beverage package recognition a challenging task.

Motivated by recent successes in sub-image retrieval [11][16], we formulate the beverage package recognition problem as a sub-image retrieval problem where two images are matched even if only *a portion* of them are similar. The query is compared to a collection of panoramic images which are unrolled and scanned beverage labels extracted from various package forms. For example, Fig. 2 (a) shows the panoramic image for a Coke can. By using panoramic images, we need only one reference image for each beverage item. The recognition of query images can then be performed by finding the most similar image in the reference dataset based on a similarity measurement that aggregates patch-to-patch similarities. Thus, two partially similar images can be considered matched.

To solve the problem where two distinct beverage packages share similar visual features, we propose a query-dependent reweighting scheme of local features to cumulate similarities between two images only from critical regions. This is derived from the observation that a keypoint's discriminative power may vary given different contexts. For example, the brand name Fanta in Fig 1 (b) is useful for recognizing Fanta series, but is not discriminative for identifying the beverage among flavor varieties in the same series.

In the remainder of the paper, we first describe related work in this field. Section 3 presents a new dataset for beverage package recognition. We then present the coarse-to-fine filtering approach for recognizing beverage packages in Section 4 and, finally, demonstrate the performance and conclude the paper with a short discussion summarizing our findings.

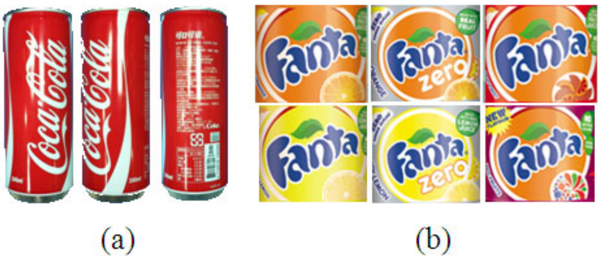


Fig. 1. Characteristics of the beverage package recognition problem. (a) Three examples of a Coke can. The most recognizable part (e.g. the brand logo) may be fully or partially captured, or totally invisible on the query image. (b) Beverages in a series: they have common visual elements in the package design.

2 Related Work

Recent works have shown some successes of product recognition using local feature based visual searches [15][17][18]. For recognizing products such as books and CD covers, there are some mobile image recognition systems on the market [1][2][3]. For local-feature-based methods, a query image is represented by a set of local features and a reference image is retrieved if it has sufficient matches of local features with the query image. More specifically, two major components—local features and image matching—are crucial to the product recognition performance.

Robust and invariant local features such as Scale-Invariant Feature Transform (SIFT) [13] and Speeded Up Robust Features (SURF) [5], are applicable for mobile search applications. These descriptors are in general high dimensional feature vectors, e.g. a SIFT feature has 128 elements. Recently, Chandrasekar *et al.* proposed the Compressed Histogram of Gradients (CHoG) [7] which captures gradient statistics from local patches in a histogram and applies tree coding techniques to compress the histograms into low bit-rate feature descriptors. In [6], an experimental study on local feature descriptors for mobile visual search compares MPEG-7 image signatures, CHoG, and SIFT and concludes that SIFT and CHoG outperform MPEG-7 image signatures greatly in terms of feature-level and image-level matching. Since our beverage package recognition system is a client-server based visual search system and the main recognition task—the computational intensive part—is performed on a server, we develop both the SIFT and SURF representations and will compare their effectiveness in the experiments.

To accelerate the matching process, the dataset is usually organized and indexed. When searching for similar instances for a query, only a small fraction of the dataset needs to be examined. Since features have a high dimensionality, classical methods such as KD-trees and its variants [8] often suffer from the “curse of dimensionality”. More recently, vector quantization and local-sensitive hashing (LSH) techniques have been popularly adopted to build a visual vocabulary of image features or to partition the feature space [15][12]. In this work, we adopt a LSH-based method [12] because of its simplicity in concept and its effectiveness for speeding up the recognition process.

3 Beverage Image Dataset

We introduce a beverage package image dataset which currently contains 60 reference images. The dataset will continue to grow. Each is a panoramic image by manually unrolled and scanned beverage labels extracted from various package forms. Figure 2 (a) shows a few examples in the dataset. The reason why we built our own dataset in this manner is because unlike other products, the beverage packages have various forms (e.g. bottles, cans, and boxes). Therefore, most of the beverage package images available on the web cannot meet our requirement, i.e. the label must be fully expanded and captured. Although the use of panoramic images compresses information of a 3D object into a 2D image, as we will shown in the experiment, the point correspondences can still be built by using robust local features. Furthermore, unlike the case in the general object recognition tasks where one category has multiple images, we require only one reference image for each beverage.

These beverage package images are essentially different from general images. The following differences are observed from our samples. Firstly, one or multiple text lines are present on the container which gives an indication related to the content of the package, such as the brand name, the product name, nutrition table, and etc. The texts are usually highlighted with a distinguishing appearance from the background. Secondly, symbolic patterns and cartoon-like figures are commonly used for the graphics design to deliver the look of freshness and delicacy. These observations are useful for identifying the visual elements for beverage package design, and upon which we illustrate why a SIFT-based representation is effective for beverage package recognition.

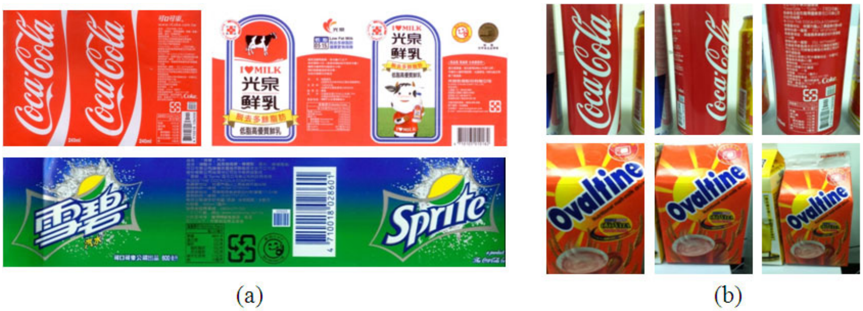


Fig. 2. (a) Three examples of our reference images. We have one reference image for each beverage; (b) Testing images. A package is captured in three different settings. Please refer to texts for details.

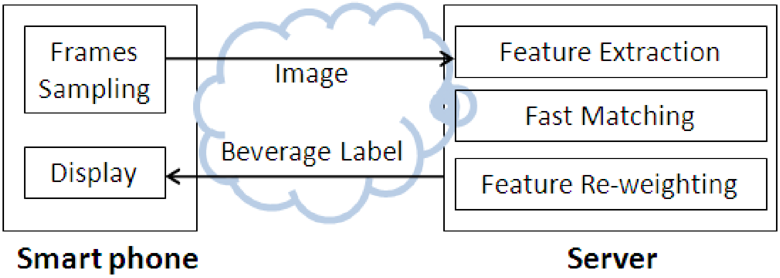


Fig. 3. The framework of our beverage package recognition system. Using a client application on a smartphone, video frames are sampled and sent to a processing server that recognizes the query image.

4 Approach

4.1 The Hierarchical Framework

Our beverage package recognition system is a client-server based visual search system as illustrated in Fig. 3. As described in Section 1, we formulate the beverage package recognition problem as a sub-image retrieval problem given a query image captured from a cell phone and a set of reference beverage package images. Since the query image may be similar to only a portion of its reference image, global feature based methods is not applicable in our application. Instead, the retrieval can be achieved by matching two images represented by local keypoints and their descriptors. Pairwise comparison among local keypoints can further measure the degree of overlapping between two images. Due to intensive computations of the feature extraction and matching processes, the recognition task is usually performed on a server.

However, not every local keypoint has equal discriminative power. For example, keypoints that capture the recycling symbol are informative, but may not be discriminative as the symbol would appear on various product containers. More importantly, *a keypoint's discriminative power may vary* given different contexts. For example, keypoints that describe the brand name "Fanta" are useful to differentiate Fanta soft drinks from others. However, these keypoints are not discriminative for identifying a particular flavor among the Fanta series. These observations are valid especially in beverage package recognition as products in the same series tend to share some common visual patterns.

Therefore, we propose a hierarchical approach which firstly performs a coarse recognition and determines the context for a refinement search. This can be achieved by using conventional keypoint matching techniques. If the coarse search returns more than on potential matches—it usually happens when the query belongs to a series, we then apply a refinement step that adjusts weights of local features under comparison to refine similarities from those matched keypoints. We now describe the approach in details.

4.2 Coarse Recognition

The first step in the hierarchical approach is designed to identify potential matches and to filter irrelevant images. Conventional image matching approaches for recognizing rigid objects (e.g. books, CD covers) [15][17][18] can be applied. For image representation, we particularly choose SIFT-like descriptors because they are constructed by summarizing the gradient information within a local region. They can capture unique edge patterns and unique local neighborhoods. These characteristics are suitable for describing symbolic patterns and cartoon-like figures which are widely used in beverage packaging design.

We now show an analysis of the SIFT and the SURF descriptor distributions of our reference image dataset using the visualization approach described in [12]. The approach aims to sketch the space of high dimensional local features by using an approximate nearest neighbor probing scheme based on 2-stable locality-sensitive hashing. Each feature is indexed to a bucket by the hashing scheme and the indexing result yields a visualization of the distribution of feature vectors—a distribution that is peaked or has a small entropy value implies that the feature is less descriptive. Figure 4 shows the SIFT and the SURF feature distributions extracted from our beverage package image dataset. The entropy values 5.9 (SIFT) and 4.18 (SURF) of the distributions indicate that both features' descriptiveness is fairly good. For example, the entropy value of the SIFT distribution extracted from the Berkeley natural images is 4.11, and the entropy value is 2.38 when SIFT features are extracted from noise patches, as reported in [12].

We identify the potential matches by examining if the number of patch correspondences between a query and a reference image exceeds a pre-defined threshold. If more than one reference images are returned in this step—mostly when the query belongs to a beverage series—we proceed to the refinement search step.

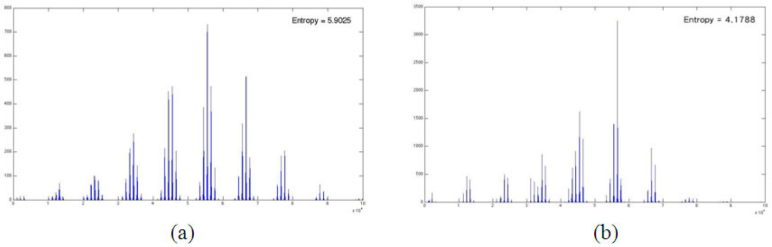


Fig. 4. The feature vector distribution over buckets: (a) SIFT (b) SURF. The entropy values 5.9 and 4.2 imply that SIFT-like local features yield a more informative feature space for beverage package images.

4.3 Refinement Search

As beverages in a series usually share common visual elements in packaging design, the similarity between the query and the candidates indexed by k returned in the previous step should be re-calculated from only the discriminative regions. We now assign a weight $w(p_i)$ to each keypoint p_i that estimates the likelihood of belonging to a particular beverage package:

$$w(p_i) = \frac{N - \sum_k t_{i,k}}{N-1}, \tag{1}$$

where N is the number of candidates, and $t_{i,k}$ is a binary variable that represents the presence of absence of a keypoint in the k-th candidate image that matches p_i. Intuitively, $w(p_i)$ relates to the occurrence frequency of p_i in candidate images. For example, $w(p_i) = 1$ if p_i is matched to keypoints that exactly appear on only one candidate image, and $w(p_i) = 0$ if all candidate images have matched keypoints. Figure 5 shows three query examples with distinct flavors. The weights of each keypoint are coded in red dots. Light color indicates more discriminative power. It is interesting to observe that most discriminative keypoints are located at their unique regions (fruit patterns in the example) while the weights of common parts, e.g. the beverage name, are assigned with a smaller value (represented with dark dots).

The weights are assigned in a similar manner to the term frequency–inverse document frequency (tf-idf) approach. However, we did not use a visual vocabulary, and, more importantly, the weights were not pre-computed from the whole reference image dataset. To save the computation, we applied a LSH based fast matching approach [12] to build the keypoint correspondences, and identified the beverage if the ratio of the top two weighted similarities is above a threshold.

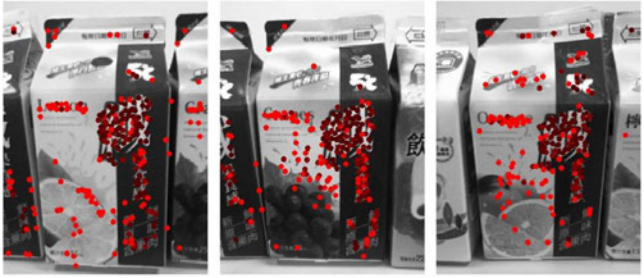


Fig. 5. Keypoints and their weights. Light color indicates more discriminative power.

5 Experiments

We present two experiments to evaluate the effectiveness of the proposed approach in recognizing beverage packages. The first experiment evaluates the overall recognition performance. To mimic the scenario how the approach will be used in practice, we captured the query images in a popular chain of convenience stores using an iPhone. Note that these query images were collected in a very different way than that of the reference images. We took three images for 40 randomly selected beverages, resulting in a testing set of 120 images. If a beverage is packed in a can or a bottle, the images were snapped with the brand name (or logo) fully, partial visible or totally invisible. For boxes, we adjusted the zoon-in factor and obtained one that contains only a portion of the box, one that captures the full box, and one that contains the box under recognition and parts of its nearby beverage packages. Figure 2 (b) illustrates a few examples.

Each image in the testing set is used as a query, and at most one beverage is retrieved. Table 1 summarizes the recognition performance. Our recipe that combines a keypoint matching method with a query-dependent weighting scheme achieves promising performance in both accuracy rate and computational speed. In particular, SIFT features achieve 15.8% higher accuracy than SURF features. We believe this is due to the fact that the manner how SURF integrates the gradient information within a patch loses some discriminative power. This leads to a worse matching result where a patch may be matched to dissimilar ones. Table 1 also lists the average runtime[1] (in seconds) for recognizing a query image. By using the LSH technique, the proposed method has a runtime of about 0.1s, and, thus, is a viable solution to applications that require real-time processing.

Figure 6 shows the images that failed in the experiment using SIFT—they are rejected by the system. As the images on the top row have a very simple design—mainly texts and color blocks—very few keypoints are available on those images. Furthermore, the camera flash creates an unnatural shininess on drink can images that may deteriorate the matching results. The number of matched keypoints thus does not exceed the threshold in the coarse recognition step. For the images on the bottom row, the system cannot differentiate the green tea and the black tea as their package designs in the "try-it" series are very similar. They differ only on portions of the

[1] The reported runtime includes all processing time between snapping a picture and the showing the relevant information on screen.

background color. Since the SIFT-based representation does not take color information into account, this difference unfortunately cannot be identified in the refinement search process.

We conducted the second experiment and examined only the beverage packages that belong to a series in order to evaluate the proposed weighting scheme. We collected additional beverage package images from the web—24 images among 9 series in which there is a corresponding reference image in our dataset, and 8 images in which there isn't. The images have 33 distinct flavors. We try to mimic the situation when a new flavor variety is launched in market while the dataset is not yet updated to include the new example. The system should have the ability to reject these queries.

Except the black tea and the green tea packages in the "try-it" line, others are correctly recognized. We examined the matching results and observed that the similarities between a query and its reference image are neatly accumulated from the discriminative regions. Furthermore, the system can successfully reject the 8 images that represent new flavor varieties. Note that the 8 images could be retrieved as a false positive by conventional systems because they have similar patterns with those packages in the same line of products. The refinement search step is essential to identify the existence of critical regions that differentiate the query from others in the same series.

The proposed system has a graphical user interface as shown in Fig. 7. It streams a video and displays frames when the application is activated. It then samples frames, performs recognition and shows relevant information if the beverage package is recognized. The phone would be used as a "scanner" for checking out product information and the usage should be easy and intuitive.

Table 1. Comparison of SIFT and SURF features for beverage package recognition. Runtime is reported in seconds.

Feature	Recognition Accuracy	Runtime (Exhaustive)	Runtime (LSH)	Speedup
SIFT (128-d)	92.5%	17.8905 (5.5385)	0.1289 (0.0368)	167.73x
SURF (64-d)	76.7%	11.0793 (2.3183)	0.1086 (0.0372)	150.50x

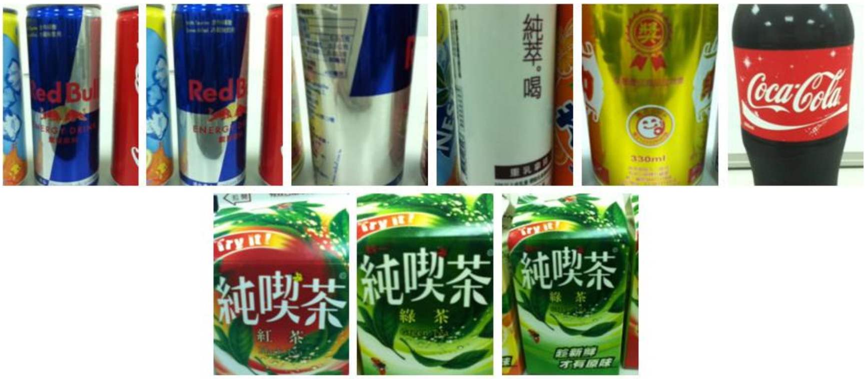
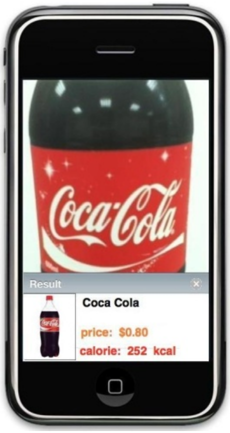


Fig. 6. Testing images for which our method failed



Coca Cola

price: $0.80

calorie: 252 kcal

Fig. 7. Example outcome of our system. The beverage is automatically recognized and annotated with product information such as price and calorie count.

6 Conclusions

In this paper we propose an approach for practical beverage package recognition for mobile application. We examine the challenges faced in the design and the development of a practical system that can achieve a fairly good recognition performance. There are a few directions we may explore to further enhance the approach. For example, the current representation is based on SIFT descriptors that describe the gray-level images alone. However, as we observed from our reference image dataset, the color design of beverage packages seems to follow certain rules. For example, the similar, contrast, or complementary hues are commonly used in the same serious of products. A representation that encodes both the shape and color information should be more effective. Furthermore, once an image is described by more than one type of descriptors, an indexing approach that can enable fast retrieval of visual instances described by multiple cues would be desired.

References

[1] Google Goggles, `http://www.google.com/mobile/googles/`

[2] SnapTell, `http://www.snaptell.com`

[3] Kooaba, `http://kooaba.com`

[4] `http://www.foodprocessing.com/wp_downloads/gt_appetite.html`

[5] Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: SURF: speeded up robust features. Computer Vision and Image Understanding (CVIU) 110(3), 346–359 (2008)

[6] Chandrasekhar, V., Chen, D., Lin, A., Takacs, G., Tsai, S., Cheung, N.-M., Reznik, Y., Grzeszczuk, R., Girod, B.: Comparison of local feature descriptors for mobile visual search. In: Proceedings of IEEE International Conference on Image Processing, Hong Kong (September 2010)

[7] Chandrasekhar, V., Takacs, G., Chen, D., Tsai, S., Grzeszczuk, R., Girod, B.: CHoG: compressed histogram of gradients. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, Miami, Florida (June 2009)

[8] Gaede, V., Gunther, O.: Multidimensional access methods. ACM Computer Survey 30(2), 170–231 (1998)

[9] Gam, H.J.: Employment of fashion orientation, shopping orientation and environmental variables to determine consumers' purchase intention of environmentally friendly clothing. In: International Textile and Apparel Association (2009)

[10] Glover, B., Bhatt, H.: RFID Essentials. O'Reilly Media (2006)

[11] Ke, Y., Sukthankar, R., Huston, L.: Efficient near-duplicate detection and sub-image retrieval. In: ACM International Conference on Multimedia, New York, NY (October 2004)

[12] Lee, W.-T., Chen, H.-T.: Probing the local-feature space of interest points. In: Proceedings of IEEE International Conference on Image Processing, Hong Kong (September 2010)

[13] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal on Computer Vision 60(2), 91–110 (2004)

[14] Palmer, R.C.: The Bar Code Book: Reading, Printing, and Specification of Bar Code Symbols, 3rd edn. Helmers Publishing (1995)

[15] Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, New York, NY (June 2006)

[16] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, Minneapolis, Minnesota (June 2007)

[17] Tsai, S., Chen, D., Singh, J., Girod, B.: Rate-efficieent, real-time CD cover recognition on a camera-phone. In: Proceedings of ACM International Conference on Multimedia, Vancouver, Canada (October 2008)

[18] Tsai, S., Chen, D., Chandrasekhar, V., Takacs, G., Cheung, N.-M., Vedantham, R., Grzeszczuk, R., Girod, B.: Mobile product recognition. In: Proceedings of ACM International Conference on Multimedia, Florence, Italy (October 2010)

An Equivalent 3D Otsu's Thresholding Method

Puthipong Sthitpattanapongsa and Thitiwan Srinark[⋆]

Graphics Innovation and Vision Engineering (GIVE) Laboratory
Department of Computer Engineering, Faculty of Engineering
Kasetsart University, Bangkok, Thailand
puthi.sthit@gmail.com, thitiwan.s@ku.ac.th

Abstract. Due to unsatisfactory segmentation results when images contain noise by the Otsu's thresholding method. Two-dimensional (2D) and three-dimensional (3D) Otsu's methods thus were proposed. These methods utilize not only grey levels of pixels but also their spatial informations such as mean and median values. The 3D Otsu's methods use both kinds of spatial information while 2D Otsu's methods use only one. Consequently the 3D Otsu's methods more resist to noise, but also require more computational time than the 2D ones. We thus propose a method to reduce computational time and still provide satisfactory results. Unlike the 3D Otsu's methods, our method selects each threshold component in the threshold vector independently instead of one threshold vector. The experimental results show that our method is more robust against noise, and its computational time is very close to that of the 2D Otsu's methods.

Keywords: Image segmentation, Thresholding, 3D Otsu's method, Three-dimensional histogram.

1 Introduction

Thresholding is considered as one low-level segmentation method since it uses only pixel information. The method is typically simple and computationally efficient. Different thresholding methods are described and compared based on different error measurements in [1]. One popular thresholding method is Otsu's [2] due to its fast computation and reasonable results in many applications. However, it uses only a one-dimensional (1D) histogram of an image, which cannot express spatial relation between image pixels, it is difficult to obtain accurate results when images contain noise. Lui *et al.* [3] thus proposed two-dimensional (2D) Otsu's method. This method selects an optimal threshold vector on a 2D histogram. The 2D histogram consists of the gray levels of the image pixels and the mean values of their neighborhood. Since the 2D histogram represents the relation of the original and mean-filtered images, this method gives more satisfactory results. However this method uses an exhaustive search to find the optimal threshold vector, the time complexity of this method is $O(L^4)$, where L

[⋆] Thanks to Kasetsart University Research and Development Institute for funding.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 358–369, 2011.

is the number of gray levels. Gong *et al.*[4] thus proposed a fast recursive method of the 2D Otsu's method which can reduce the time complexity from $O(L^4)$ to $O(L^2)$. Ningbo *et al.* [5] proposed a method, which projects a 2D histogram onto a diagonal line to compose a new 1D histogram. The method uses a 1D Otsu's method to select a point that splits this histogram into object and background regions, and applies a 2D Otsu's method to select an optimal threshold vector. This method can enhance execution time, but it requires a large space for three look-up tables. Yue *et al.* [6] proposed a decomposition of the 2D Otsu's method that calculates the optimal threshold by using two 1D Otsu's computations instead of one 2D Otsu's computation. This method is robust against noise, and the time complexity is reduced from $O(L^2)$ to $O(L)$. Chen *et al.* [7] pointed out the weakness of region division by a threshold vector in the 2D Otsu's method that some object and background regions are assigned to edge and noise regions, and vice versa. They proposed the 2D Otsu's method on a gray level-gradient histogram, however, an appropriate initialization is required.

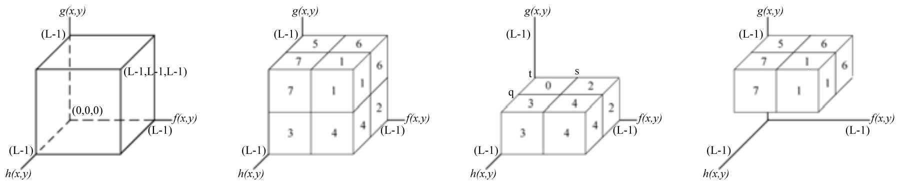
In addition to 2D Otsu's methods, Jing *et al.* [8] proposed a three-dimensional (3D) Otsu's method that selects an optimal threshold vector on a 3D histogram. This 3D histogram contains the median values of neighborhood pixels as the third feature. The 3D Otsu's method provides better results than the 2D Otsu's methods, but its time complexity is $O(L^3)$. Wang *et al.* [9] proposed a group of new recurrence formula of the 3D Otsu's method. This method thus removes redundant computation and calculates a look-up table by iteration. The method has the same thesholding results as the traditional 3D Otsu's method, however, its time complexity is still $O(L^3)$. Dongju *et al.* [10] proved that the objective function of K-means is equivalent to that of the Otsu's method, K-means thus can be extended to 2D and 3D thesholding methods. and performs more efficiently than Otsu's.

Notice that the time complexity of the 2D's Otsu methods can be reduced from $O(L^4)$ to $O(L)$ while the time complexity of the 3D Otsu's methods is still at $O(L^3)$. Even though K-means can be used instead of Otsu's methods, its execution time depends on the number of iterations. In this paper, we propose a fast and robust thresholding method, which selects and uses three optimal thresholds independently instead of one threshold vector of 3D's Otsu methods. Our method can reduce the time complexity from $O(L^3)$ to $O(L)$, and it still provides satisfactory results in noisy conditions.

2 3D Otsu's Method

Given an image $f(x, y)$ represented by L gray levels and the number of pixels in the image, N. The mean and the median of gray values of pixels in the $k \times k$ neighborhood regions centered at the coordinate (x, y) are denoted as $g(x, y)$ and $h(x, y)$, respectively, which are defined as

$$g(x,y) = \frac{1}{k^2} \sum_{i=-k/2}^{k/2} \sum_{j=-k/2}^{k/2} f(x+i, y+j) \tag{1}$$



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

Fig. 1. Three-dimensional histogram

$$h(x,y) = med\left\{ f(x+i, y+j) : i = -\frac{k}{2}, \ldots, \frac{k}{2};\ j = -\frac{k}{2}, \ldots, \frac{k}{2} \right\} \tag{2}$$

In this paper, we use $k = 3$. For each pixel in the image, we can obtain a triple (i, j, k), where i is the original gray level appeared in $f(x, y)$, j is the grey level of the mean value appeared in $g(x, y)$, and k is the gray level of the median value appeared in $h(x, y)$. All the triples of the image define a 3D histogram within a cube of $L \times L \times L$ as shown in Fig.1(a). Let c_{ijk} denote the frequency of a triple (i, j, k). Its joint probability can be expressed as

$$p_{ijk} = \frac{c_{ijk}}{N}, \tag{3}$$

where $0 \leq i, j, k \leq L-1$ and $\sum_i^{L-1} \sum_j^{L-1} \sum_k^{L-1} p_{ijk} = 1$

Given an arbitrary threshold vector (s, t, q). This threshold vector divides the 3D histogram into eight rectangular volumes as shown in Fig. 1(b)-1(d). Let C_0 and C_1 represent the object and the background, respectively, or vice versa; m_x, ω_x, and μ_x represent the summation vector, the probability, and the mean vector of the rectangular volume x (R_x), respectively, where x is the rectangular volume number; and μ_T represent the total mean vector. m_x can be expressed as

$$\begin{aligned} m_x &= \omega_x \mu_x = (m_{xi}, m_{xj}, m_{xk})^T \\ &= \left(\sum_{(i,j,k) \in R_x} i p_{ijk}, \sum_{(i,j,k) \in R_x} j p_{ijk}, \sum_{(i,j,k) \in R_x} k p_{ijk} \right)^T \end{aligned} \tag{4}$$

The three elements in the triple are very close to each other for the interior pixels of either the object or the background regions while they are very different for the pixels that are edges and noise. Therefore, the rectangular volumes 2-7 can be considered as noise and edges; and rectangular volumes 0 and 1 can be considered as object and background regions, respectively, or vice versa. In most cases, the edge and noise pixels are very small fraction of the overall pixels in an image, hence the probabilities of the rectangular volumes 2-7 can be negligible. It can easily verify the relations,

$$\omega_0 + \omega_1 \approx 1 \qquad \omega_0 \mu_0 + \omega_1 \mu_1 \approx \mu_T \tag{5}$$

The probabilities of C_0 and C_1 thus can be denoted as

$$\omega_0 = \sum_{(i,j,k)\in R_0} p_{ijk} = \sum_{i=0}^{s}\sum_{j=0}^{t}\sum_{k=0}^{q} p_{ijk} \tag{6}$$

$$\omega_1 = \sum_{(i,j,k)\in R_1} p_{ijk} = \sum_{i=s+1}^{L-1}\sum_{j=t+1}^{L-1}\sum_{k=q+1}^{L-1} p_{ijk} \tag{7}$$

The mean vectors of C_0 and C_1 can be expressed as

$$\begin{aligned}\mu_0 = (\mu_{0i}, \mu_{0j}, \mu_{0k})^T &= \left(\frac{m_{0i}}{\omega_0}, \frac{m_{0j}}{\omega_0}, \frac{m_{0k}}{\omega_0}\right)^T \\ &= \left(\sum_{(i,j,k)\in R_0} \frac{ip_{ijk}}{\omega_0}, \sum_{(i,j,k)\in R_0} \frac{jp_{ijk}}{\omega_0}, \sum_{(i,j,k)\in R_0} \frac{kp_{ijk}}{\omega_0}\right)^T \end{aligned} \tag{8}$$

$$\begin{aligned}\mu_1 = (\mu_{1i}, \mu_{1j}, \mu_{1k})^T &= \left(\frac{m_{1i}}{\omega_1}, \frac{m_{1j}}{\omega_1}, \frac{m_{1k}}{\omega_1}\right)^T \\ &= \left(\sum_{(i,j,k)\in R_1} \frac{ip_{ijk}}{\omega_1}, \sum_{(i,j,k)\in R_1} \frac{jp_{ijk}}{\omega_1}, \sum_{(i,j,k)\in R_1} \frac{kp_{ijk}}{\omega_1}\right)^T \end{aligned} \tag{9}$$

The total mean vector of 3D histogram is

$$\begin{aligned}\mu_T &= (\mu_{iT}, \mu_{jT}, \mu_{kT})^T \\ &= \left(\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\sum_{k=0}^{L-1} ip_{ijk}, \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\sum_{k=0}^{L-1} jp_{ijk}, \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\sum_{k=0}^{L-1} kp_{ijk}\right)^T \end{aligned} \tag{10}$$

The between-class discrete matrix is defined as

$$S_B(s,t,q) = \omega_0[(\mu_0 - \mu_T)(\mu_0 - \mu_T)^T] + \omega_1[(\mu_1 - \mu_T)(\mu_1 - \mu_T)^T] \tag{11}$$

The trace of discrete matrix can be expressed as

$$\begin{aligned} tr(S_B(s,t,q)) = \omega_0[(\mu_{0i} - \mu_{Ti})^2 + (\mu_{0j} - \mu_{Tj})^2 + (\mu_{0k} - \mu_{Tk})^2] + \\ \omega_1[(\mu_{1i} - \mu_{Ti})^2 + (\mu_{1j} - \mu_{Tj})^2 + (\mu_{1k} - \mu_{Tk})^2] \end{aligned} \tag{12}$$

The optimal threshold vector (s', t', q') is

$$(s', t', q') = arg \max_{0\le s,t,q\le L-1} (tr(S_B(s,t,q))) \tag{13}$$

3 Proposed Method

From (5), we can see that $\omega_x \approx 0$ and $\omega_x \mu_x = m_x \approx 0$, where $x = 2, \ldots, 7$. From these conditions, we can conclude as follows.

$$\omega_0 \approx \omega_{0i} = \omega_0 + \omega_3 + \omega_5 + \omega_7 = \sum_{i=0}^{s} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} p_{ijk} = \sum_{i=0}^{s} P_i \tag{14}$$

$$\omega_1 \approx \omega_{1i} = \omega_1 + \omega_2 + \omega_4 + \omega_6 = \sum_{i=s+1}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} p_{ijk} = \sum_{i=s+1}^{L-1} P_i \tag{15}$$

$$\omega_0 \approx \omega_{0j} = \omega_0 + \omega_2 + \omega_3 + \omega_4 = \sum_{i=0}^{L-1} \sum_{j=0}^{t} \sum_{k=0}^{L-1} p_{ijk} = \sum_{j=0}^{t} P_j \tag{16}$$

$$\omega_1 \approx \omega_{1j} = \omega_1 + \omega_5 + \omega_6 + \omega_7 = \sum_{i=0}^{L-1} \sum_{j=t+1}^{L-1} \sum_{k=0}^{L-1} p_{ijk} = \sum_{j=t+1}^{L-1} P_j \tag{17}$$

$$\omega_0 \approx \omega_{0k} = \omega_0 + \omega_2 + \omega_5 + \omega_6 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{q} p_{ijk} = \sum_{k=0}^{q} P_k \tag{18}$$

$$\omega_1 \approx \omega_{1k} = \omega_1 + \omega_3 + \omega_4 + \omega_7 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=q+1}^{L-1} p_{ijk} = \sum_{k=q+1}^{L-1} P_k \tag{19}$$

$$m'_{0i} = m_{0i} + m_{3i} + m_{5i} + m_{7i} = \sum_{i=0}^{s} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} i p_{ijk} = \sum_{i=0}^{s} i P_i \tag{20}$$

$$m'_{1i} = m_{1i} + m_{2i} + m_{4i} + m_{6i} = \sum_{i=s+1}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} i p_{ijk} = \sum_{i=s+1}^{L-1} i P_i \tag{21}$$

$$m'_{0j} = m_{0j} + m_{2j} + m_{3j} + m_{4j} = \sum_{i=0}^{L-1} \sum_{j=0}^{t} \sum_{k=0}^{L-1} j p_{ijk} = \sum_{j=0}^{t} j P_j \tag{22}$$

$$m'_{1j} = m_{1j} + m_{5j} + m_{6j} + m_{7j} = \sum_{i=0}^{L-1} \sum_{j=t+1}^{L-1} \sum_{k=0}^{L-1} j p_{ijk} = \sum_{j=t+1}^{L-1} j P_j \tag{23}$$

$$m'_{0k} = m_{0k} + m_{2k} + m_{5k} + m_{6k} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{q} k p_{ijk} = \sum_{k=0}^{q} k P_k \tag{24}$$

$$m'_{1k} = m_{1k} + m_{3k} + m_{4k} + m_{7k} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=q+1}^{L-1} k p_{ijk} = \sum_{k=q+1}^{L-1} k P_k \tag{25}$$

where

$$m_{0i} \approx m'_{0i}, m_{1i} \approx m'_{1i}, m_{0j} \approx m'_{0j}, m_{1j} \approx m'_{1j}, m_{0k} \approx m'_{0k}, m_{1k} \approx m'_{1k}.$$

Thus, we can define the new mean vectors as

$$\begin{aligned}\mu_0 \approx \mu'_0 = (\mu'_{0i}, \mu'_{0j}, \mu'_{0k})^T &= \left(\frac{m'_{0i}}{w_{0i}}, \frac{m'_{0j}}{w_{0j}}, \frac{m'_{0k}}{w_{0k}}\right)^T \\ &= \left(\frac{\sum_{i=0}^{s} iP_i}{\sum_{i=0}^{s} P_i}, \frac{\sum_{j=0}^{t} jP_j}{\sum_{j=0}^{t} P_j}, \frac{\sum_{k=0}^{q} kP_k}{\sum_{k=0}^{q} P_k}\right)^T \qquad (26)\end{aligned}$$

$$\begin{aligned}\mu_1 \approx \mu'_1 = (\mu'_{1i}, \mu'_{1j}, \mu'_{1k})^T &= \left(\frac{m'_{1i}}{w_{1i}}, \frac{m'_{1j}}{w_{1j}}, \frac{m'_{1k}}{w_{1k}}\right)^T \\ &= \left(\frac{\sum_{i=s+1}^{L-1} iP_i}{\sum_{i=s+1}^{L-1} P_i}, \frac{\sum_{j=t+1}^{L-1} jP_j}{\sum_{j=t+1}^{L-1} P_j}, \frac{\sum_{k=q+1}^{L-1} kP_k}{\sum_{k=q+1}^{L-1} P_k}\right)^T \qquad (27)\end{aligned}$$

where $P_i = \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} p_{ijk}$, $P_j = \sum_{i=0}^{L-1} \sum_{k=0}^{L-1} p_{ijk}$, and $P_k = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ijk}$. Notice that P_i, P_j, and P_k are equivalent with the 1D histogram of of original, mean-filtered, and median-filtered images, respectively. From (14)-(19) and (26)-(27), we can rewritten (12) as

$$\begin{aligned}tr(S_B(s,t,q)) \approx\ &\overbrace{[\omega_{0i}(\mu'_{0i} - \mu_{Ti})^2 + \omega_{1i}(\mu'_{1i} - \mu_{Ti})^2]}^{A} + \\ &\overbrace{[\omega_{0j}(\mu'_{0j} - \mu_{Tj})^2 + \omega_{1j}(\mu'_{1j} - \mu_{Tj})^2]}^{B} + \qquad (28)\\ &\overbrace{[\omega_{0k}(\mu'_{0k} - \mu_{Tk})^2 + \omega_{1k}(\mu'_{1k} - \mu_{Tk})^2]}^{C}\end{aligned}$$

The values of terms A, B, and C depend on the values of s, t, and q, respectively. We can define each term as

$$\sigma_{Bi}(s) = \omega_{0i}(\mu'_{0i} - \mu_{Ti})^2 + \omega_{1i}(\mu'_{1i} - \mu_{Ti})^2 \qquad (29)$$

$$\sigma_{Bj}(t) = \omega_{0j}(\mu'_{0j} - \mu_{Tj})^2 + \omega_{1j}(\mu'_{1j} - \mu_{Tj})^2 \qquad (30)$$

$$\sigma_{Bk}(q) = \omega_{0k}(\mu'_{0k} - \mu_{Tk})^2 + \omega_{1k}(\mu'_{1k} - \mu_{Tk})^2 \qquad (31)$$

The optimal threshold (s', t', q') is

$$\begin{aligned}(s', t', q') &= arg \max_{0 \le s,t,q \le L-1} (tr(S_B(s,t,q))) \\ &\approx arg \max_{0 \le s,t,q \le L-1} (\sigma_{Bi}(s) + \sigma_{Bj}(t) + \sigma_{Bk}(q)) \qquad (32)\end{aligned}$$

which can be splited into

$$s' = arg \max_{0 \le s \le L-1} \sigma_{Bi}(s) \qquad (33)$$

$$t' = arg \max_{0 \le t \le L-1} \sigma_{Bj}(t) \qquad (34)$$

$$q' = arg \max_{0 \le q \le L-1} \sigma_{Bk}(q) \qquad (35)$$

Equations (33), (34), and (35) are 1D Otsu's methods that select the optimal threshold of the original, mean-filtered, and median-filtered images, respectively. Notice that we select the optimal threshold from three 1D histograms instead of one 3D histogram. Therefore, the time complexity of this method is only $O(L)$ instead of $O(L^3)$. We then apply each threshold element as a classifier to classify each image pixel into either the object or the background independently. A pixel (x, y) is assigned to the class, which is mostly selected by the thresholds s', t', and q' in the original, mean-filtered, and median-filtered images, respectively.

4 Experimental Results

We performed all experiments on a personal computer with 2.0 GHz Intel(R) Core(TM)2 Duo CPU and 4 GB DDR II memory. We implemented the proposed method in Visual C++ with OpenCV. Scilab was used to generate noised added images for noise tolerant tests. We tested on two kinds of noise including Salt&Pepper noise and Gaussian noise. Salt&Pepper noise is represented by noise density (δ), the probability of swapping a pixel. Gaussian noise is represented by mean (μ) and variance (σ^2). In our experiments, we used only $\mu = 0$.

We compared our method with the 1D Otsu's method [2], Gong's method [4] as the 2D Otsu's method, Wang's method [9] as the 3D Otsu's method, K-means [10] based methods for both 2D and 3D ones, Ningbo's method [5], and Yue's method [6] because they are based on Otsu's. For each experiment that the ground truth is available, we use misclassification error (ME) to present the number of background pixels wrongly assigned to the foreground, and vice versa; and we use modified Hausdorff distance (MHD) to measure the shape distortion of each result image compared with its corresponding ground truth. ME and MHD are defined as [1]

$$\mathsf{ME} = 1 - \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O| + |F_O|}, \tag{36}$$

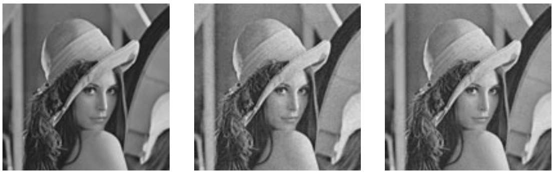
$$\mathsf{MHD} = max(d_{\mathsf{MHD}}(F_O, F_T), d_{\mathsf{MHD}}(F_T, F_O)), \tag{37}$$

where

$$d_{\mathsf{MHD}}(F_O, F_T) = \frac{1}{|F_O|} \sum_{f_O \in F_O} \min_{f_T \in F_T} \|f_O - f_T\|,$$

$$d_{\mathsf{MHD}}(F_T, F_O) = \frac{1}{|F_T|} \sum_{f_T \in F_T} \min_{f_O \in F_O} \|f_T - f_O\|.$$

F_i and B_i denote the foreground and background pixels, respectively, of an image i, which includes the ground truth (O) and thresholded (T) images. $|.|$ is the cardinality of the set. $\|f_O - f_T\|$ is the Euclidean distance between the two corresponding pixels of the ground truth and thresholded images. Notice that ME varies from 0 (a perfectly classified image) to 1 (a totally incorrect binarized image).



(a) Original (b) Gaussian (c) Salt&Pepper

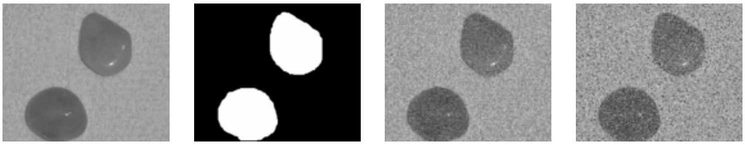
Fig. 2. Lena images w/o noise added

Table 1. Optimal thresholds of Lena images w/o noise added

Methods	Fig.		
	2(a)	2(b)	2(c)
1D Otsu's	117	119	117
2D Otsu's	(123,117)	(123,126)	(117,192)
3D Otsu's	(130,125,117)	(132,122,121)	(130,126,117)
2D K-means	(117,117)	(118,118)	(117,117)
3D K-means	(117,117,117)	(118,118,118)	(117,117,117)
Ningbo's	(117,117)	(117,119)	(117,117)
Yue's	(117,117)	(119,118)	(117,117)
Proposed	(117,117,117)	(117,117,117)	(117,117,117)

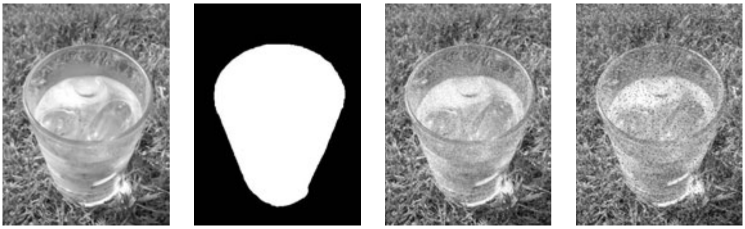
In the first experiment, we compared the optimal threshold selected by each method. We segmented Lena images consisting of the original one, and two noise added images. The first noise added image was generated by adding Salt&Pepper noise with $\delta = 0.01$ to the original image, and the other one was generated by adding Gaussian noise with $\sigma^2 = 0.005$ to the original image as shown in Fig. 2. Optimal thresholds are shown in Table 1. It can be seen that the optimal threshold of the proposed method is close to the optimal threshold of the other methods.

In the second experiment, we tested the robustness of each method in the presence of noise. We selected two images as our test images from *Segmentation evaluation database*[11]. Fig. 3(a) and 3(b) show the first test image and its ground truth, respectively. Fig. 4(a) and 4(b) show the second test image and its ground truth, respectively. We added noise to each test image to generate new 51 images with Salt&Pepper noise using δ that are vary from 0 to 0.1, and the other 51 images with Gaussian noise using σ^2 that are vary from 0 to 0.01. Fig. 3(c) and 3(d) show example noise added images of the first test image. Fig. 4(c) and 4(d) show example noise added images of the second test image. Both test images show difficulties for thresholding when some amount of noise is added. Fig. 5(a) shows the histogram of the first test image that clearly presents bimodal, while Fig. 5(c) shows the histogram of the second test image that does not clearly presents bimodal. Fig. 5(b) and 5(d) show histograms of two images with Gaussian noise added. Both of them present a single modal with a jagged curve. The second test image itself can be challenged to segment such that some background pixels present similar gray levels as the object. We



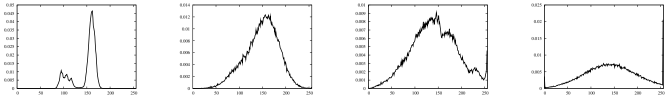
(a) Original (b) Ground truth (c) Gaussian ($\sigma^2 = 0.01$) (d) Salt&Pepper ($\delta = 0.1$)

Fig. 3. The first image set with sample noise added images in the second experiment



(a) Original (b) Ground truth (c) Gaussian ($\sigma^2 = 0.01$) (d) Salt&Pepper ($\delta = 0.1$)

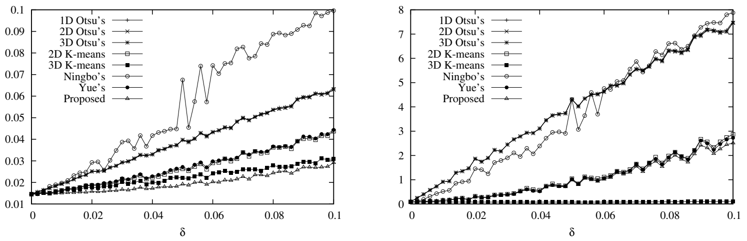
Fig. 4. The second image set with sample noise added images in the second experiment



(a) Histogram of Fig.3(a) (b) Histogram of Fig.3(c) (c) Histogram of Fig.4(a) (d) Histogram of Fig.4(c)

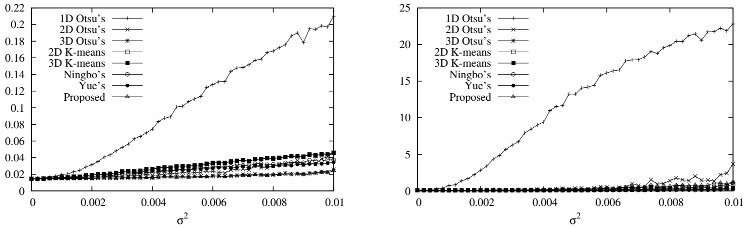
Fig. 5. Histograms of the test images in the second experiment

segmented these 204 noise added images. We evaluated the performance of each method based on ME and MHD. Fig. 6 and 7 show the evaluation results of the first test images. Fig. 8 and 9 show the evaluation results of the second test images. From the evaluation results in the presence of Salt&Pepper noise shown in Fig. 6 and 8, both ME and MHD values of our method are lower than those of the other methods except MHD values on the first test images, MHD values of our method are higher than the 3D K-means method. The thresholding results of the 3D K-means and our methods are shown in Fig. 10. The 3D K-means method gives higher number of mistaken pixels in the object region, and lower number of mistaken pixels in the background region, however, our method gives lower number of mistaken pixels in the object region, and higher number of pixels in the background region. MHD of our method is thus higher than of the 3D K-means. From the evaluation results in the presence of Gaussian noise shown in Fig. 7 and 9, both ME and MHD values of our method on the second test images are lower than those of the other methods. ME values of our method on the first test images are very close to that of the 3D Otsu's method and lower than those of the other methods. MHD values of our method on the first test images



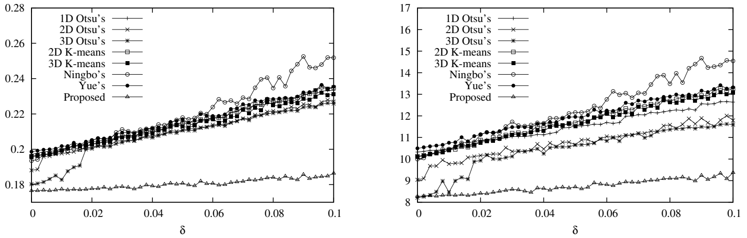
(a) ME (b) MHD

Fig. 6. Comparison of ME and MHD for thresholding of the first test images with Salt&Pepper noise added at various δ in the second experiment



(a) ME (b) MHD

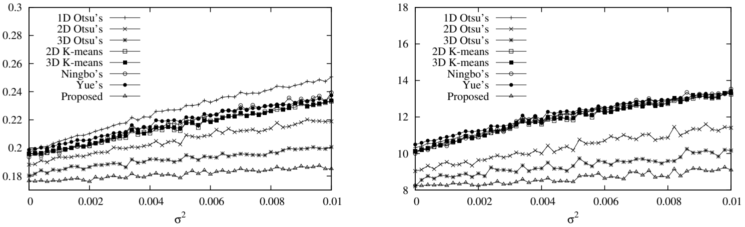
Fig. 7. Comparison of ME and MHD for thresholding of the first test images with Gaussian noise added at various σ^2 in the second experiment



(a) ME (b) MHD

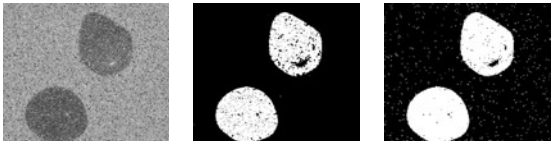
Fig. 8. Comparison of ME and MHD for thresholding of the second test images with Salt&Pepper noise added at various δ in the second experiment

are a little higher than those of the other methods except the 1D Otsu's and 2D Otsu's methods. The average computational time on all noise added images are 0.08, 12.05, 1891.63, 27.21, 1259.54, 11.39, 6.29, and 13.19 ms, for the 1D Otsu's, 2D Otsu's, 3D Otsu's, 2D K-means, 3D K-means, Ningbo's, Yue's, and our proposed methods, respectively. It can be seen that our method performs faster than the other 3D methods. Our average execution time is nearly the same as that of the other 2D methods except Yue's method. Our method always gives low error measurements in both classification and shape evaluations.



(a) ME (b) MHD

Fig. 9. Comparison of ME and MHD for thresholding of the second test images with Gaussian noise added at various σ^2 in the second experiment



(a) Original (b) 3D K-means (c) proposed

Fig. 10. The original image and thresholded images of 3D K-means and proposed methods when σ^2=0.01

In the last experiment, we tested our method and the others with 200 real images from the *Segmentation evaluation database* [11], where the ground truth of each image is provided. The average error measurements ($\overline{\mathsf{ME}}$ and $\overline{\mathsf{MHD}}$) and the average compuational time ($\overline{\mathsf{T}}$) over 200 test images of each method are shown in Table 2. Segmentation results can be seen at `http://give.cpe.ku.ac.th/thresholding/equivalent-3D-thresholding.php`. From the results, it can be seen that the average computational time of our method is lower than that of the other 3D methods and is almost the same as that of the other 2D methods except the Yue's method. The average ME and MHD values of our method is lower than that of the other methods. It indicates that our method shows the best matching of the object and the background, and also gives the smallest amount of shape distortion.

Table 2. $\overline{\mathsf{ME}}$, $\overline{\mathsf{MHD}}$, and $\overline{\mathsf{T}}$ over 200 real images

Method	$\overline{\mathsf{ME}}$	$\overline{\mathsf{MHD}}$	$\overline{\mathsf{T}}$ (ms)
1D Otsu's	0.217102	19.545244	0.32
2D Otsu's	0.214391	19.580417	11.99
3D Otsu's	0.213022	19.569091	2151.03
2D K-means	0.228411	19.603090	20.60
3D K-means	0.228340	19.616353	1027.13
Ningbo's	0.214194	19.627783	12.92
Yue's	0.214962	19.408089	6.66
Proposed	0.211341	19.199144	12.59

5 Conclusions

We presented an improved thresholding method to overcome the shortcoming of the 1D, 2D, and 3D Otsu's method. The method calculates each optimal threshold from the original, mean-filtered, and median-filtered images independently; and uses the most selected class by each threshold on the corresponding images as the thresholding results. We tested our method on real images and images with noise added. The results show that our method gives satisfactory results, and it is robust against noise. Moreover, it requires less computational time than the other 3D methods, and also gives better or comparable results.

References

1. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. Jour. of Electronic Imaging 13(1), 146–168 (2004)
2. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. on Systems, Man and Cybernetics 9(1), 62–66 (1979)
3. Liu, J., Li, W., Tian, Y.: Automatic thresholding of gray-level pictures using two-dimension otsu method. In: Proc. of Intl. Conf. on Circuits and Systems, China, vol. 1, pp. 325–327 (1991)
4. Gong, J., Li, L., Chen, W.: Fast recursive algorithms for two-dimensional thresholding. Pattern Recognition 31(3), 295–300 (1998)
5. Ningbo, Z., Gang, W., Gaobo, Y., Weiming, D.: A fast 2d otsu thresholding algorithm based on improved histogram. In: Chinese Conf. on Pattern Recognition (CCPR), pp. 1–5 (2009)
6. Yue, F., Zuo, W.M., Wang, K.Q.: Decomposition based two-dimensional threshold algorithm for gray images. Zidonghua Xuebao/Acta Automatica Sinica 35(7), 1022–1027 (2009)
7. Chen, Y., Chen, D.r., Li, Y., Chen, L.: Otsu's thresholding method based on gray level-gradient two-dimensional histogram. In: 2nd Intl. Asia Conf. on Informatics in Control, Automation and Robotics (CAR), vol. 3, pp. 282–285 (2010)
8. Jing, X.J., Li, J.F., Liu, Y.L.: Image segmentation based on 3-d maximum between-cluster variance. Tien Tzu Hsueh Pao/Acta Electronica Sinica 31(9), 1281–1285 (2003)
9. Wang, L., Duan, H., Wang, J.: A fast algorithm for three-dimensional otsu's thresholding method. In: IEEE Intl. Sym. on IT in Medicine and Education (ITME), pp. 136–140 (2008)
10. Dongju, L., Jian, Y.: Otsu method and k-means. In: Ninth Intl. Conf. on Hybrid Intelligent Systems (HIS), vol. 1, pp. 344–349 (2009)
11. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)

Human Motion Tracking with Monocular Video by Introducing a Graph Structure into Gaussian Process Dynamical Models

Jianfeng Xu, Koichi Takagi, and Shigeyuki Sakazawa

KDDI R&D Laboratories Inc.
{ji-xu,ko-takagi,sakazawa}@kddilabs.jp

Abstract. This paper presents a novel approach to tracking articulated human motion with monocular video. In a conventional tracking system based on particle filters, it is very challenging to track a complex human pose with many degrees of freedom. A typical solution to this problem is to track the pose in a low dimensional latent space by manifold learning techniques, e.g., the Gaussian process dynamical model (GPDM model). In this paper, we extend the GPDM model into a graph structure (called *GPDM graph*) to better express the diverse dynamics of human motion, where multiple latent spaces are constructed and dynamically connected to each other appropriately by an unsupervised learning method. Basically, the proposed model has both intra-transitions (in each latent space) and inter-transitions (among latent spaces). Moreover, the probability of inter-transition is dynamic, depending on the current latent state. Using the proposed GPDM graph model, we can track human motion with monocular video, where the average tracking errors are improved from the state-of-the-art methods in our experiments.

Keywords: motion tracking, monocular video, manifold learning, Gaussian process dynamical model, motion graph.

1 Introduction

In the computer vision community, much effort has been put into inferring the human pose or 3D articulated human body parts from videos [6,11]. Basically, there are two kinds of approaches: on one side, *discriminative approaches* employ a parametric model mapping directly from image observation to the pose space [1,15]. Although recent techniques are developed with promising performance [15], it is generally quite difficult to learn of such mapping because the mapping itself is generally ambiguous, e.g. two different poses may have almost the same observation. On the other side, the inverse problem of generating image observations by a given pose is well defined, leading the *generative approaches* to optimize the pose (or pose distribution). As a typical technique of generative approaches, particle filters are widely adopted to track human motion from videos [7,13,16] and are also employed in this paper.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 370–383, 2011.

In most papers [1,7,13,14,15,16], the human pose is represented as articulated human body parts in a tree structure with many degrees of freedom [6,11]. Therefore, the human pose is very difficult to track directly in the high dimensional pose space due to the curse of dimensionality with such techniques as the particle filters [7]. Fortunately, recent studies demonstrate that human motion can essentially be described in a much lower dimensional space (called *latent space*) [9,16,17], given that targeted motion has regular dynamics. In this paper, the Gaussian process dynamical models (GPDM) proposed by Wang et al. [17] is employed because of the good performance as reported by Quirion et al. [12] for many applications in tracking human motion [7,16]. However, it is unsatisfactory for a single GPDM to express complicated motion that has several motion patterns [7].

Our basic idea is to separate complicated motion into simple segments, where a GPDM model (i.e., latent space) is learned for each segment. Naturally, those latent spaces should be transited with a probability. Moreover, the transition probability among latent spaces (called *inter-transition*) should depend on the current state of the current latent space. For example, the probability of inter-transitions is much higher at the landing state than that at the flight state from the jumping space to walking space. Generally speaking, it is very challenging to learn such a complicated latent model in a reasonable way. For this purpose, we combine the techniques of the motion graph [3,8,10] and GPDM to construct our novel model *GPDM graph*. As far as we know, it is the first latent dynamics model with a graph structure. In addition, our approach is a completely unsupervised learning method by the data-driven scheme.

Although monocular approaches are much more challenging than multi-view approaches due to incomplete information, such as the occlusion problem [6,11], a single camera is more ubiquitous and cheaper, thus making it suitable for non-professional users. Moreover, a single camera solution can open up a new possibility to capture motion from video archives such as past Olympic games. In both cases, currently, we do not require real time processing, targeting to the applications for entertainment, coaching, etc.

The rest of this paper is organized as follows. Section 2 presents a brief survey on related work. Section 3 describes the proposed algorithm in detail. Section 4 discusses our experimental results on the HumanEva dataset [13]. The conclusions and future work are addressed in section 5.

2 Related Work

A plethora of literature is reported on video-based human motion tracking. See the comprehensive reviews in previous surveys [6,11]. In this section, we focus on dimension reduction and particle filter techniques for human motion tracking, which are the categories of our core techniques.

Although principle component analysis (PCA) is widely used for dimension reduction in human motion [2], linear mapping has poor ability to reduce the dimensions because human motion is highly non-linear. As a non-linear approach, the Gaussian process latent variable model (GPLVM) can learn the latent space and the mapping function [9]. GPLVM is an efficient tool for modeling distribution in a high dimensional space with a compact low dimensional representation. Wang et al. extend GPLVM to GPDM [17], which models the dynamics in the learned latent space. GPDM and its variants, including BGPDM [16], are widely employed in tracking human motion because it simultaneously models the latent space, the dynamics in the latent space, and the mapping from latent space to the pose space. GPDM is an unsupervised method and only needs a minimum of learning data [17]. However, Chen et al. [7] have reported that GPDM cannot model complicated motion. They introduce a switching GPDM model that is successfully used in human motion tracking [7]. In their model, the transition probability of switching states is static. Moreover, labels of switching states in the learning data are usually required, which means that it is a supervised learning method. The essential difference between our GPDM graph and the switching GPDM is whether to learn *dynamic* switching probability with an *unsupervised* method, which is very challenging but important in real applications.

On the other hand, particle filters and variants are successfully applied to track objects in video because of the compatibility of non-linear and non-Gaussian elements [4]. However, the workable dimensionality for particle filters is small as pointed out by Chen et al. [7]. With the above dimension reduction methods, it is possible to track human motion using particle filters in a low dimensional latent space. In this paper, we employ a particle filter technique similar to Sigal et al. [13]. Our experimental results show that performance is further improved from the state-of-the-art methods [7,14,17]. See the details in Section 4.

3 Proposed Method

Our system includes learning the GPDM graph and inference with GPDM graph. To learn the GPDM graph, training motion data are divided into several short segments, and a GPDM model is simultaneously learned for each segment. At the same time, the candidates for inter-transitions among GPDM models are detected using the short-term principle component analysis, originally proposed by Xu et al. [19]. With the learned GPDM graph, which includes the mapping function from latent space to pose space, the human pose is inferred with the low dimensional latent space by particle filters. In this stage, inter-transitions are dynamically determined by the similarity of human poses. In Section 3.1, we will first describe the concept of the GPDM graph in detail.

3.1 Concept of GPDM Graph

The basic hypothesis is that a complicated motion consists of a sequence of elemental motions, and each elemental motion, originally in many degrees of freedom, is essentially controlled by low dimensional latent space as shown in Eq. (2) [7,17]. At the same time, the first-order Markov dynamics is assumed for simplicity in latent spaces as shown in Eq. (1). Furthermore, we connect the latent spaces with a dynamic probability as shown in Fig. 1 (called *inter-transitions*).

$$\mathbf{z}_t^k = f(\mathbf{z}_{t-1}^k; \mathbf{A}) + \mathbf{n}_{z,t} \tag{1}$$

$$\mathbf{x}_t = g(\mathbf{z}_t^k; \mathbf{B}) + \mathbf{n}_{x,t} \tag{2}$$

where $\mathbf{z}_t^k \in \mathbb{R}^d$ denotes the d-dimensional coordinates at time-t in the k-th latent space, $\mathbf{x}_t \in \mathbb{R}^D$ denotes the D-dimensional coordinates at time-t in pose space ($D >> d$), f and g are non-linear mappings parameterized by $\mathbf{A}$ and $\mathbf{B}$, and $\mathbf{n}_{z,t}$ or $\mathbf{n}_{x,t}$ denotes zero-mean, isotropic, white Gaussian noise processes. Note that our model has multiple latent spaces but a single pose space while the original GPDM has a single latent space and a single pose space. Therefore, our model is more general and suitable for complex motions.

One of the unique characteristics in our model is that the inter-transition probability depends on the current state of the current latent space, which infers that probability changes dynamically. The example in Fig. 1 explains the reasonableness of our model, where two kinds of elemental motions exist including "walking" and "jumping". As shown in Fig. 1(b), it is natural that the transition probability at the landing state is much higher than at the flight state when transiting from "jumping" to "walking". Similarly, the transition probability must be dynamic when transiting from "walking" to "jumping" as shown in Fig. 1(c). Surely, besides the inter-transitions, we have intra-transitions in each latent space as the original GPDM did [17]. Note that our model is designed for not only the above scenario that clearly has two motions but also the complex motion with multiple short phases that can transit in-between such as the gesture motion in Table 2.

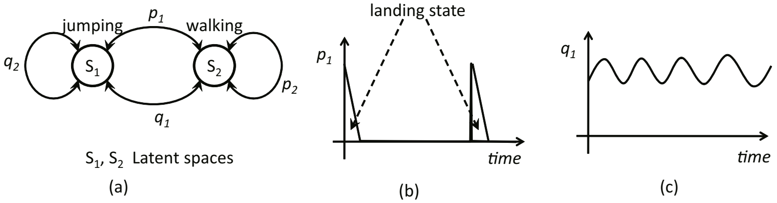


Fig. 1. Concept of the proposed GPDM graph model (a): multiple latent spaces are connected in a probability depending on the current state of the current latent space. Naturally, the probability of inter-transition is much higher at the landing state than that at the flight state from the jumping space to walking space in (b). Similarly, the transition probability is dynamic when transiting from walking to jumping in (c).

Specifically, when using GPDM to learn latent space, the above model can be further represented as Eqs. (3) and (4) through Gaussian process regression, where the dynamics of the latent space is the former, and the mapping from latent space to pose space is the latter. Note that both are probability functions, which are desirable for particle filters. For more details, please refer to [17].

$$p(\mathbf{Z}^k \mid \bar{\alpha}^k) = \frac{p(\mathbf{z}_1^k)}{\sqrt{(2\pi)^{(N-1)d} \mid \mathbf{K}_{Zk} \mid^d}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_{Zk}^{-1}\mathbf{Z}_{\mathbf{2:N}}{}^k\mathbf{Z}_{2:N}^{k\,T})\right) \quad (3)$$

$$p(\mathbf{X} \mid \mathbf{Z}^k, \bar{\beta}^k, \mathbf{W}^k) = \frac{\mid \mathbf{W}^{\mathbf{k}} \mid^N}{\sqrt{(2\pi)^{ND} \mid \mathbf{K}_{Xk} \mid^D}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_{Xk}^{-1}\mathbf{X}\mathbf{W}^{\mathbf{k}^2}\mathbf{X}^T)\right) \quad (4)$$

where $\mathbf{Z}^k \equiv \mathbf{Z}_{1:N}^k \equiv \{\mathbf{z}_1^k, \mathbf{z}_2^k, ..., \mathbf{z}_N^k\}$ denotes all the coordinates in the k-th latent space, $\mathbf{X} \equiv \{\mathbf{x}_t : t = 1, ..., N\}$ denotes all coordinates in the pose space, $\bar{\alpha}^k$ denotes kernel hyperparameter vector for dynamics in latent space, which is used in calculating the kernel function $(\mathbf{K}_{Zk})_{ij} \equiv k_{Zk}(\mathbf{z}_i^k, \mathbf{z}_j^k)$ in Eq. (5), $\bar{\beta}^k$ and $\mathbf{W}^k \equiv diag(w_1^k, ..., w_D^k)$ are hyperparameters for the mapping function, where the kernel function $(\mathbf{K}_{Xk})_{ij} \equiv k_{Xk}(\mathbf{x}_i, \mathbf{x}_j)$ is calculated by Eq. (6). In a word, a GPDM model is represented as $\{\mathbf{Z}^k, \bar{\alpha}^k, \bar{\beta}^k, \mathbf{W}^k\}$, which is learned in a segment of motion data.

$$k_{Zk}(\mathbf{z}_i^k, \mathbf{z}_j^k) = \exp\left(-\frac{\beta_1^k}{2} \parallel \mathbf{z}_i^k - \mathbf{z}_j^k \parallel^2\right) + (\beta_2^k)^{-1}\delta_{\mathbf{z}_i^k, \mathbf{z}_j^k} \quad (5)$$

$$k_{Xk}(\mathbf{x}_i, \mathbf{x}_j) = \alpha_1^k \exp\left(-\frac{\alpha_2^k}{2} \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2\right) + \alpha_3^k \mathbf{x}_i^T \mathbf{x}_j + (\alpha_4^k)^{-1}\delta_{\mathbf{x}_i, \mathbf{x}_j} \quad (6)$$

The probability of an inter-transition is intuitively calculated according to the distance between two poses that are transited as Eq. (7), where the principle in the so-called motion graph technique is adopted [3,8,10]. Basically, the more similar the poses are, the higher the transition probability is.

$$-\log p(\mathbf{z}_t^k \rightarrow \mathbf{z}_{t'}^{k'}) \propto dist(\mathbf{x}_t, \mathbf{x}_{t'}) \quad (7)$$

where $\mathbf{z}_t^k$ denotes the departure coordinates in the k-th latent space, where the mean of the mapping function is $\mathbf{x}_t$ in the pose space, and $\mathbf{z}_{t'}^{k'}$ denotes the destination coordinates in the k'-th latent space, where the mean of the mapping function is $\mathbf{x}_{t'}$ in the pose space. The function *dist* is a distance function between two poses. See an implementation by Wang et al. [18], where the weighted difference of joint orientations is calculated as Eq. (8).

$$dist(\mathbf{x}_t, \mathbf{x}_{t'}) = \sum_{n=1}^{m} w_k \parallel \log(q_{t',n}^{-1} q_{t,n}) \parallel^2) \quad (8)$$

where m denotes the number of joints in the human pose, and $q_{t,n}$ denotes the orientation of joint n in the t-th frame, expressed as quaternion.

3.2 Learning of GPDM Graph

Given human motion, the proposed GPDM graph will be learned with an unsupervised method.

Inter-transition Candidate Detector: It is necessary to detect the possible inter-transitions in the training motion data, e.g. the time instants for hitting the ground in walking motion, where a short-term principal component analysis (short-term PCA) method [19] is employed. The basic idea in short-term PCA is piece-wise linear approximation for non-linear human motion because motion data are almost linear in the short term due to strong temporal coherence. Short-term PCA is executed in a sliding window in the joint position space. And the peaks and valleys of the coordinates in the first principal component are regarded as candidates for inter-transitions $\{\mathbf{b}_i : i = 1, ..., I\}$. The detected candidates for inter-transitions are stored as potential time instants to transit to other motions. See the detailed procedure in [19].

Construction of GPDM Graph: We simultaneously segment training motion data and learn a sequence of GPDM models. The basic idea is to use the trial and error approach iteratively with a sliding window as shown in Table 1. The motion in a window is called a *motion clip*, which is empirically set as 60 frames or 0.5 seconds in our implementation. We merge the motion clips when the reconstruction error, calculated as Eq. (9), is smaller than the threshold as shown in Table 1. Here, the threshold is set as 1.0. Otherwise, it is divided into two segments at the boundary of additional motion clip as shown in Fig. 5(b). In concept, a segment for a motion pattern is desired. In practice, the real concern in the inference is the reconstruction error.

$$error(t) = dist(\mathbf{x}_t, \hat{\mathbf{x}}_t) \tag{9}$$

$$\hat{\mathbf{x}}_t = g(\hat{\mathbf{z}}_\mathbf{t}^\mathbf{k}) \tag{10}$$

where $\hat{\mathbf{x}}_t$ is the t-th reconstructed pose from the t-th coordinates $\hat{\mathbf{z}}_t^k$ of a so called *mean prediction sequence* in the current latent space, generated from $\mathbf{z}_1^k$ by simulating the dynamical process one frame at a time [17].

Now, our GPDM graph is composed of the GPDM models $\{\mathbf{Z}^k, \bar{\alpha}^k, \bar{\beta}^k, \mathbf{W}^k : k = 1, 2, ..., K\}$ and all the candidates for inter-transitions $\{\mathbf{b}_i : i = 1, ..., I\}$, which will be used in the next section. An example is shown in Fig. 2, where a walking motion in Section 4 is used.

Table 1. Procedure for learning a sequence of GPDM models

```
while training data are not finished
    do add a motion clip
        merge the current clip
        learn the GPDM for merged motion
        if $error(t) < TH$ for any $t$
            then continue
        else learn the GPDM without the added clip
            output the learned GPDM
            reset the start point as the head of current clip
            break
```

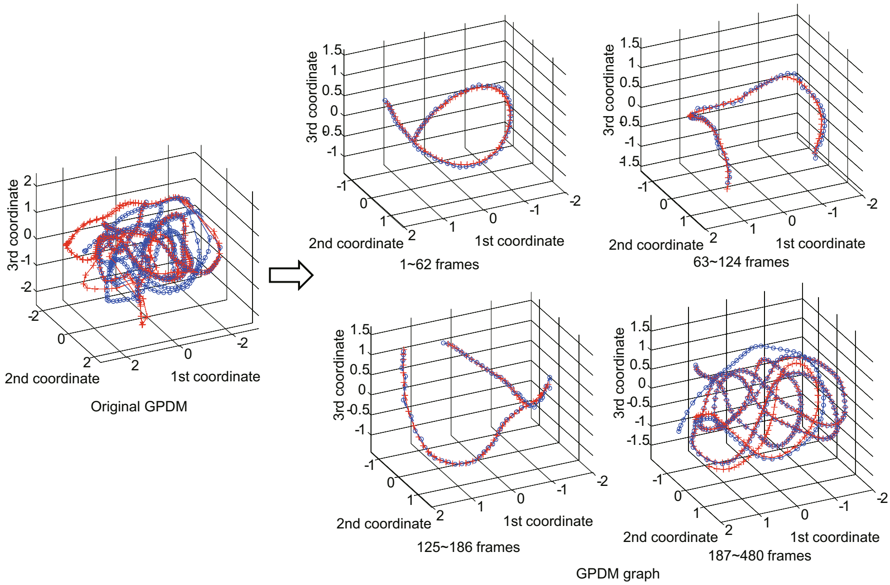


Fig. 2. An example of the proposed GPDM graph model for a walking motion in Table 2, where the circles denote the learned coordinates in latent space and the crosses denote the predicted coordinates in latent space.

3.3 Inference with GPDM Graph

As mentioned before, particle filters are used to infer the human pose from the input video, where the main difference from conventional particle filters [13] is that the particles are generated in the latent spaces instead of the pose space, reducing the space dimension greatly. Later, the particles in the latent space are called *latent particles* $\mathbf{z}_t^k(1 : P^k)$, which denotes the P^k coordinates in the k-th latent space for time t. The corresponding particles in the pose space are called *pose particles* $\mathbf{x}_t(1 : P)$, which denotes the $P(= \sum P^k)$ coordinates in the pose space for time t and is calculated by the mean of GP regression in Eq. (6) as Wang et al. [17] reported.

Similar to conventional particle filters [13], the initialization is specially processed. In detail, the ground truth of the first frame is used to generate particles. First, we search the human poses in the training motion data to find pose candidates, which are required to be similar to the first frame (i.e. satisfied by Eq. (11)). The corresponding coordinates in the learned latent space are the seeds for latent particles $\mathbf{z}_1^k(1 : P^c)$ with P^c particles. P^c is determined by Eqs. (12)-(14) given P particles in total. With the seeds and particle number, the latent particles $\mathbf{z}_1^k(1 : P^c)$ for the first frame are generated by a Gaussian distribution. Those latent particles are further mapped to the pose particles $\mathbf{x}_1(1 : P)$ ($P = \sum P^c$ is the total particle number). The importance weights $w_t(1 : P)$ are equally set as $1/P$.

$$dist(\mathbf{x}_1^{gt}, \mathbf{x}_t^*) < \overline{dist} \text{ and } \frac{d(dist(\mathbf{x}_1^{gt}, \mathbf{x}_t^*))}{dt} < 0 \tag{11}$$

$$-\log q(c) = dist(\mathbf{x}_1^{gt}, \mathbf{x}_t^*(c)) / \sum_i dist(\mathbf{x}_1^{gt}, \mathbf{x}_t^*(i)) \tag{12}$$

$$p(c) = q(c) / \sum_i q(i) \tag{13}$$

$$P^c = p(c) * P \tag{14}$$

where $\mathbf{x}_1^{gt}$ denotes the ground truth of the first frame, $\mathbf{x}_t^*(c)$ denotes a pose candidate, and $\overline{dist}$ denotes the average distance for all the pose candidates.

Then, the human pose is inferred by the following steps iteratively. Note that this scheme can easily be extended to variants of the particle filters, such as the annealed particle filter [13].

1. **Likelihood calculation:** With the pose particles $\mathbf{x}_t(1:P)$ and video frame $\mathbf{y}_t$, the importance weights $\hat{w}_t(1:P)$ are updated by the same likelihood functions as [13], which includes the edge and silhouette features in the video frame.
2. **Resampling:** According to the updated importance weights, resample the latent particles $\hat{\mathbf{z}}_t^k(1:P^k)$, which is similar to [13].
3. **Prediction by inter-transition:** This step is unique for our GPDM graph model. The above latent particles are checked whether they should be transited to other latent spaces. By this step, the particles are adaptively distributed among the latent spaces. Since all possible inter-transitions are learned in section 3.2, the distances are calculated between $\{\mathbf{b}_i : i = 1, ..., I\}$ and each pose particle $\mathbf{x}_t(p)$, which is mapped from a latent particle $\hat{\mathbf{z}}_t^k(p)$. If the distance with $\mathbf{b}_i$ and $\mathbf{x}_t(p)$ is smaller than the threshold, the latent particle $\hat{\mathbf{z}}_t^k(p)$ will be transited to the k'-th latent space corresponding to the human pose $\mathbf{b}_i$. The transited particle number is determined by the distances and the original particle number, which is similar to Eq. (14).
4. **Prediction by intra-transition:** Although this step exists in conventional particle filters, much more advanced dynamics is available in the latent space using GPDM models [7,16]. The purpose of this step is to generate latent particles at the next time instant $\mathbf{z}_{t+1}^k(1:P^k)$, which is calculated by the learned dynamics in Eq. (5).
5. **Mapping to pose particles:** With the above latent particles $\mathbf{z}_{t+1}^k(1:P^k)$, the pose particles $\mathbf{x}_{t+1}(1:P)$ are obtained by the mapping function in Eq. (6). Now go to Step (1) for tracking human pose in the next frame.

4 Experimental Results

Experimental Conditions: In Section 4, we evaluate our algorithm in both the learning and inference stages using the HumanEva dataset [13], where the training and test data from S1 subject are used as shown in Table 2.

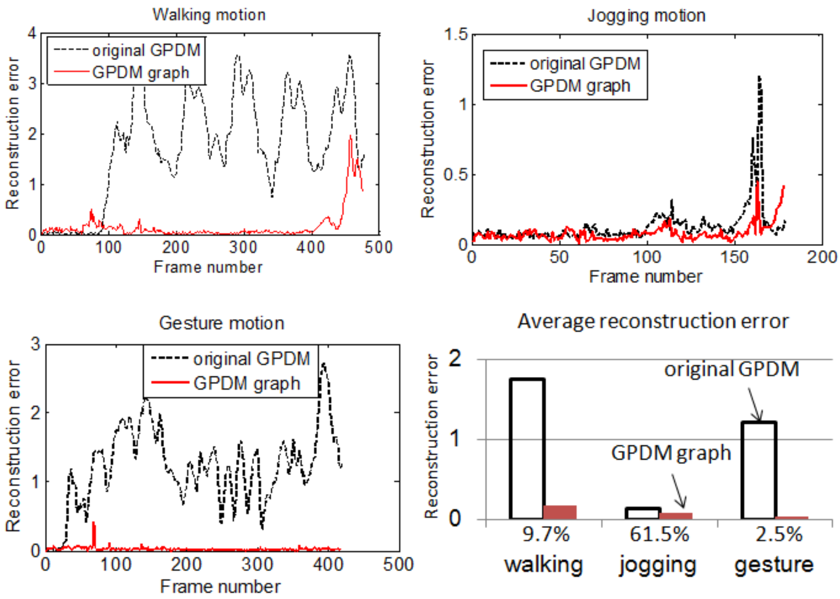


Fig. 3. Comparison of reconstruction error

Evaluation of GPDM Graph Learning: We compare our GPDM graph with the original GPDM model [17] for the three motions in Table 2. The reconstruction errors are shown in Fig. 3, where the average errors are reduced to 9.7%, 61.5%, and 2.5% in the three motions, respectively. As expected, the model precision is much improved. Basically, the more complex the motion is, e.g. gesture motion, the more benefit the proposed method provides.

Figure 4 shows the inter-transition candidates for training data. The frame distance, which means the probability of inter-transition in our method, changes a lot in Fig. 4, requiring that the transition probability should dynamically depend on the current state. Similar results were reported in motion graph technique [3,8,10]. At the same time, the inter-transition candidates should locate the similar poses with short distances in those cyclic motions. The experiments show our inter-transition candidate detector works well, which detects the local extreme values by short-term PCA [19] as shown by the crosses in Fig. 4.

Table 2. Experimental data used in the learning and inference stages

motion	description	training data	test data (C1 camera)
walking	cyclic motion	1∼480	481∼600
jogging	cyclic motion	1∼180	531∼650
gesture	multiple patterns	1∼420	421∼570

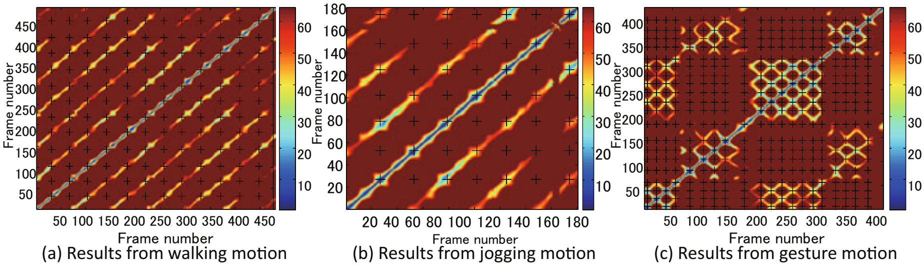


Fig. 4. Frame distances and detected candidates for inter-transitions from a walking motion (a), a jogging motion (b), and a gesture motion (c). Blue color denotes the low distance and deep red color denotes the high distance. Crosses denote the detected candidates for inter-transitions.

An interesting observation from Figs. 2 and 3 is that there are multiple patterns in a semantically simple walking motion. This is due to the following fact that the signals in two cycles are rather different. Figure 5 (a) shows the learned latent space from the first two cycles (frame #1∼#150) of the walking motion in Table 2. It is clear that the predicted latent coordinates (crosses, generated by the GPDM model) are almost the same in two cycles while the learned latent coordinates (circles, learned directly from the training data) are quite different, which infers that the learned GPDM model cannot confidently generate correct latent coordinates and leads to the reconstruction errors become rather large in the second cycle as shown in Fig. 5(b). By segmenting into two models, the reconstruction ability is greatly improved as shown in Fig. 2.

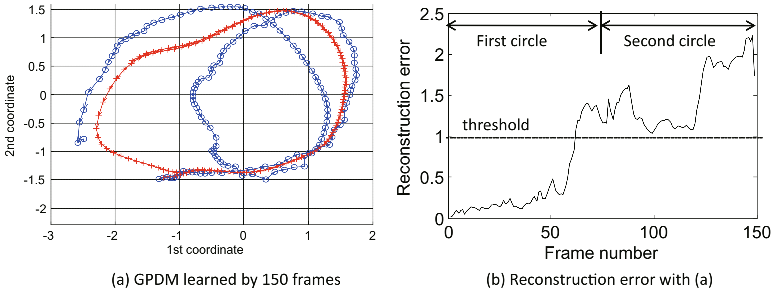


Fig. 5. Learned latent space from frame #1∼#150 of the walking motion (about two cycles), where the circles denote the learned coordinates in latent space and the crosses denote the predicted coordinates in latent space. A single GPDM model may fail to model a semantically simple motion.

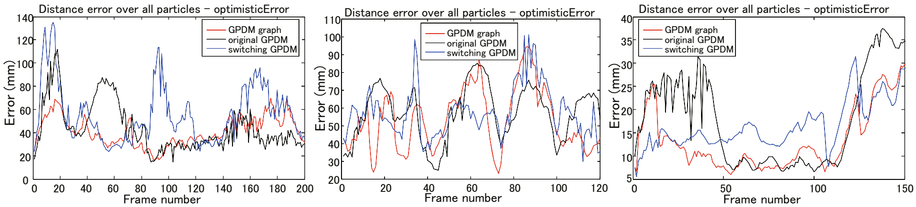
Evaluation of Pose Inference: We compare the GPDM graph model with the original GPDM model [17] and the switching GPDM model [7] where the probability of inter-transitions is constant (i.e. independent of the latent state

Table 3. Average errors of tracking human motion by different methods

motion	original GPDM	switching GPDM	GPDM graph
walking	44.16 mm	56.09 mm	40.88 mm
jogging	57.21 mm	57.55 mm	53.26 mm
gesture	17.80 mm	16.33 mm	13.23 mm

in GPDMs). In all the methods, the total particle number is set as 1000. For evaluation, the tracking error is calculated by the inferred pose and the ground truth as described by Sigal et al. [13].

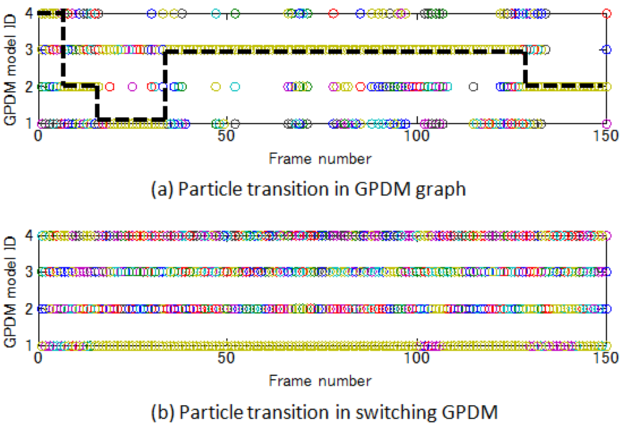
Figure 6 shows the tracking errors of the above three methods respectively, whose average error is listed in Table 3. In the above experiments, the proposed GPDM method achieves the best performance by combining the merits of original GPDM and switching GPDM[1]. As Fig. 6 shows, the GPDM graph method basically has the errors similar to the lower ones of the original GPDM and the switching GPDM. When the motion is in a single pattern, the particles in particle filter are preferred to stay in a GPDM model. On the other hand, when the motion transits to a new pattern, the particles are preferred to transit to another GPDM model. Our experimental results infer that neither the original GPDM nor the switching GPDM deals with the situations well. In this meaning, by the adaptive probability of inter-transitions, the efficiency of using particles in the particle filter is improved in the proposed GPDM graph model, leading to better performance. Figure 7 shows the particles are transited among different GPDM models by GPDM graph and switching GPDM respectively. As the dashed line in Fig. 7 (a) shows, the particles are transited properly with the motion patterns in GPDM graph while they are equally transited in switching GPDM as Fig. 7 (b) shows. Basically, in the proposed GPDM graph, the particles can automatically follow the changes of motion patterns by the adaptive transition probability among different GPDM models, which is the essential advantage of our method.



(a) Tracking error of a walking motion
(b) Tracking error of a jogging motion
(c) Tracking error of a gesture motion

Fig. 6. Tracking errors by the proposed GPDM graph (red dotted curves), the original GPDM (black solid curves), and the switching GPDM (blue dashed curves) in a walking video (a), a jogging video (b), and gesture video (c) of the S1 subject from the C1 camera

[1] As a latest result on walking motion of S1 subject in HumanEva dataset, Taylor et al. reported an average error of 47.29 mm by a sixth-order model of Implicit Mixture of Conditional Restricted Boltzmann Machines in a similar condition [14].

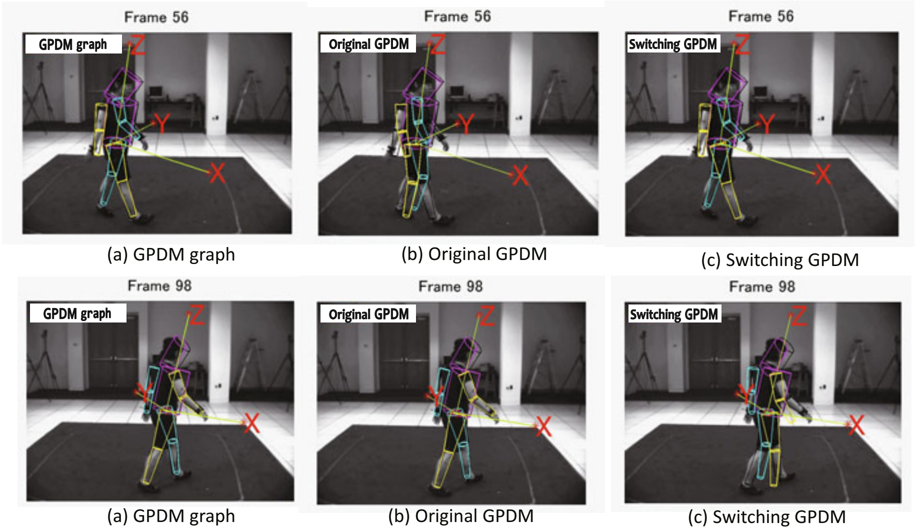


(a) Particle transition in GPDM graph

(b) Particle transition in switching GPDM

Fig. 7. Particle transitions among different GPDM models by GPDM graph (a) and switching GPDM (b) from the gesture motion. The dashed line shows the transition trace of most particles in GPDM graph. The particles are transited properly with the motion patterns in GPDM graph (a) while they are equally transited in switching GPDM (b). The color of points denotes the particle ID.

Finally, we show two samples in Fig. 8 where the proposed method tracks the pose correctly while other methods may fail to track the legs.



(a) GPDM graph (b) Original GPDM (c) Switching GPDM

(a) GPDM graph (b) Original GPDM (c) Switching GPDM

Fig. 8. Tracking result of frame #56 and #98 by GPDM graph (a), original GPDM (b), and switching GPDM (c) in the test video of walking motion. The colored cylinders show the tracking results and the black cylinders denote the ground truth.

5 Conclusions and Future Work

In this paper, our main contribution is to propose a novel model for tracking human motion from a monocular video, where the novelties are as follows.

– It is the first latent dynamics model with graph structure. With inter-transitions in the graph, the long-term correlation is possible to be used. We simultaneously segment the training motion and learn the GPDM models by the trial and error approach.
– Our data-driven approach is a completely unsupervised learning method. For this purpose, we employ the short-term PCA method to search the candidates for inter-transitions. In the inference stage, the connections (inter-transitions) are dynamically determined by the similarity of human poses, which is inspired by the motion graph technique [3,8,10].

In the future, we plan to improve the likelihood function in the tracking stage using more advanced features, such as robust local and global appearance features [5,16].

Acknowledgment. The HumanEva dataset and baseline codes are provided by Brown University. Part of GPDM codes are downloaded from Toronto University.

References

1. Agarwal, A., Triggs, B.: Recovering 3d human pose from monocular images. IEEE Trans. on PAMI 28(1), 44–58 (2006)
2. Arikan, O.: Compression of motion capture databases. ACM Trans. on Graphics 25(3), 890–897 (2006)
3. Arikan, O., Forsyth, D.A.: Interactive motion generation from examples. ACM Trans. on Graphics 21(3), 483–490 (2002)
4. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Trans. on Signal Processing 50(2), 174–188 (2002)
5. Balan, A., Black, M.J.: An adaptive appearancemodel approach formodel-based articulated object tracking. In: IEEE CVPR, vol. 1, pp. 758–765 (2006)
6. Moeslund, T.B., Hilton, A., Kruger, V.: A survey of advances in vision based human motion capture and analysis. CVIU 104(2), 90–126 (2006)
7. Chen, J., Kim, M., Wang, Y., Ji, Q.: Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition. In: IEEE CVPR, pp. 2655–2662 (2009)
8. Kovar, L., Gleicherl, M., Pighin, F.: Motion graphs. ACM Trans. on Graphics 21(3), 473–482 (2002)
9. Lawrence, N.: Gaussian process latent variable models for visualization. In: Proc. Adv. Neural Inf. Process. pp. 329–336 (2003)
10. Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S.: Interactive control of avatars animated with human motion data. ACM Trans. on Graphics 21(3), 491–500 (2002)

11. Poppe, R.: Vision-based human motion analysis: an overview. CVIU 108(1/2), 4–18 (2007)
12. Quirion, S., Duchesne, C., Laurendeau, D., Marchand, M.: Comparing gplvm approaches for dimensionality reduction in character animation. Journal of WSCG 16(1-3), 41–48 (2008)
13. Sigal, L., Balan, A., Black, M.J.: Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. International Journal of Computer Vision 87(1), 4–27 (2010)
14. Taylor, G.W., Sigal, L., Fleet, D.J., Hinton, G.E.: Dynamical binary latent variable models for 3d human pose tracking. In: IEEE CVPR, pp. 631–638 (2010)
15. Urtasun, R., Darrell, T.: Sparse probabilistic regression for activity independent human pose inference. In: IEEE CVPR, pp. 1–8 (2008)
16. Urtasun, R., Fleet, D., Fua, P.: 3d people tracking with gaussian process dynamical models. In: IEEE CVPR, pp. 238–245 (2006)
17. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Trans. on PAMI 30(2), 283–298 (2008)
18. Wang, J., Bodenheimer, B.: Synthesis and evaluation of linear motion transitions. ACM Trans. on Graphics 27(1), 1:1–1:15 (2008)
19. Xu, J., Takagi, K., Yoneyama, A.: Beat induction from motion capture data using short-term principal component analysis. The Journal of The Institute of Image Information and Television Engineers 64(4), 577–583 (2010)

Depth Map Up-Sampling Using Random Walk

Gyo-Yoon Lee and Yo-Sung Ho

Gwnagju Institute of Science and Technology (GIST),
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712 Republic of Korea
{gyoyoon,hoyo}@gist.ac.kr

Abstract. For the high quality three-dimensional broadcasting, depth maps are important data. Although commercially available depth cameras capture high-accuracy depth maps in real time, their resolutions are much smaller than those of the corresponding color images due to technical limitations. In this paper, we propose the depth map up-sampling method using a high-resolution color image and a low-resolution depth map. The proposed method is appropriate to match boundaries between the color image and the depth map. Experimental results show that our method enhances the depth map resolution successfully.

Keywords: 3D broadcasting, multi-view video, FTV, TOF cameras, depth map, interpolation, random walk.

1 Introduction

Three-dimensional (3-D) video currently attracts public attention in a variety of multimedia applications. The current 3-D videos provide 3-D effects using the stereoscopic images which are based on binocular depth cues. In the near future, users will be able choose their viewpoints of themselves in the immersive visual scenes created by 3-D videos.

Since we cannot transmit videos of all viewpoints, we synthesize the viewpoint's video using transmitted video-plus-depth data for 3-D TV [1]. To provide the high quality synthesized view, accurate depth information is important. In general, depth estimation methods are classified into two categories: active depth estimation and passive depth estimation. The active depth estimation method directly obtains the depth map using physical sensors. On the contrary, the passive depth estimation method calculates the depth values using acquired 2-D images.

The passive depth estimation method uses two or more 2-D images. Typical examples are shape from focus [2] and stereo matching [3]. The passive depth estimation method can be performed at a low price because it needs only 2-D images. However it does not guarantee quality of depth map because the performance of passive depth estimation depends on image properties. Active depth estimation uses the physical sensor such as lasers, infrared rays (IR), or light patterns. There are structured light patterns [4] and depth cameras [5]. If we use physical equipment, we can obtain more accurate depth values. However, depth cameras are expensive and they capture low-resolution depth maps only. To get accurate depth maps, the hybrid

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 384–394, 2011.

camera system was proposed [6]. To overcome problems of previous depth estimation methods, the hybrid camera system consists of the multi-view color cameras and the depth cameras. Thus, it can perform both active depth estimation and passive depth estimation.

Although depth estimation methods have been researched continually, more accurate depth estimation method remains an unsolved problem. We obtain more accurate depth values when we use the active depth estimation method. However, we need the up-sampling process due to the difference between color images and depth maps. Besides, depth up-sampling can be used for the depth encoding algorithm. We improve coding efficiency by transmitting the down-sampled depth map which is represented much fewer bits than that of original depth information. In the decoder, the transmitted depth map can be used through the up-sampling process. Thus if accuracy of up-sampled depth values is higher, coding efficiency will be improved. In this paper, we propose the efficient depth map up-sampling method.

Section 2 explains the previous depth up-sampling methods. In Section 3, we describe the proposed up-sampling method. Then, Section 4 demonstrates the experimental results. We conclude in Section 5.

2 Related Works

The hybrid camera system also has the problem that resolution of depth maps captured by depth cameras is smaller than that of the corresponding color images due to technical limitations of the depth cameras [7]. Figure 1 shows the resolution difference between the color image and the depth map of the hybrid camera system.

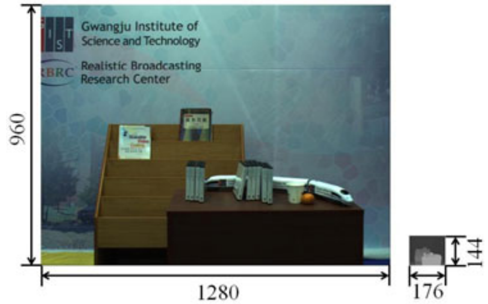

Gwangju Institute of Science and Technology
Realistic Broadcasting Research Center
960
1280
144
176

Fig. 1. Resolution of the color image and the depth map

Since the inaccurate depth information deteriorates the quality of synthesized views and 3-D video, the quality of depth maps is very important for an image-based rendering. Thus the accurate enhancement method of the low-resolution depth map is required. To obtain accurate depth information, we need to keep following properties of depth maps.

1. Boundaries of depth maps match corresponding color image boundaries.
2. Depth values of neighboring pixels in the same object are similar.

To solve this problem, various methods have been proposed. In the beginning of the research, general image interpolation methods were used such as bilinear, nearest-neghbor, and bicubic interpolations [8]. However, they do not guarantee the depth map properties. So, the Markov random field probability model and the bilateral filter are proposed.

2.1 Markov Random Field

Diebel *et al.* interpolated depth values using the Markov random field probability model (MRF) and the designed the adaptive weighting function according to the color gradient [9]. The MRF is composed of 5 node types. Figure 2 shows the designed MRF.

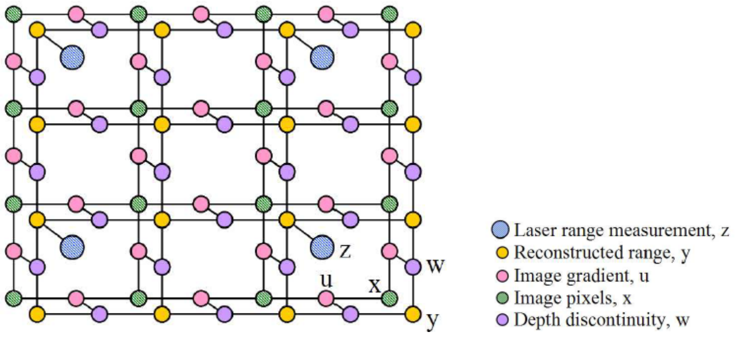


Fig. 2. Node types of MRF

The MRF is defined through thefollowing conditional probability.

$$p(y|x,z) = \frac{1}{Z}\exp(-\frac{1}{2}(\Psi + \Phi)) \tag{1}$$

where Ψ is depth measurement potential and it represents the difference between the laser range measurement and the reconstructed range. Φ is depth smoothness term. As computing the optimization problem of Eq. (1), we obtain the depth values.

2.2 Joint Bilateral Up-Sampling

Kopf *et al.* proposed the post-processing step using the bilateral filter [10]. The bilateral filter is an edge-preserving filter. The idea is to apply a spatial filter to the low resolution S, while similar range filter is jointly applied on the full resolution image. The up-sampled solution $\tilde{S}_p$ is then obtained as

$$\tilde{S}_P = \frac{1}{k_p} \sum_{q_\downarrow \in \Omega} S_{q\downarrow} f(\|p_\downarrow - q_\downarrow\|) g(\|\tilde{I}_p - \tilde{I}_q\|) \tag{2}$$

$g(\|\tilde{I}_p - \tilde{I}_q\|)$ is color distances between pixel p and q in full resolution $\tilde{I}$. And $f(\|p_{\downarrow} - q_{\downarrow}\|)$ represents the spatial distance.

After releasing the joint bilateral filter, many up-sampling methods which use modified bilateral filter are proposed. Yang *et al.* proposed the post-processing step using the bilateral filter [11]. This method enhances the low-resolution depth map by refining iteratively initial depth values.

3 Proposed Depth Up-Sampling

We generate the new depth value using the initial depth values. We warp the pixel from low-resolution depth map to color image. We define the initial values as warped depth values.

3.1 Initial Value

Camera Calibration. To match a depth map and a color image of different cameras, it is important to find out relative camera information through camera calibration [12]. We apply a camera calibration algorithm [13] to each camera and obtain projection matrices.

$$P = K[R|t]\ . \tag{3}$$

where *P* is the projection matrix of each camera. It is consist of the intrinsic matrix *K*, the rotation matrix *R*, and translation vector *t*.

3-D Warping. The camera parameter represents the relative position of the camera and world coordinates. Since we have position information and depth information of cameras, we can find the any position of the depth map in the world coordinate using Eq. (4) which is consist of camera parameter ***R, K, t***.

$$\mathrm{X_r = R_r^{-1} \cdot K_r^{-1} \cdot x_r \cdot d_r(x_r) - R_r^{-1} \cdot t_r}\ . \tag{4}$$

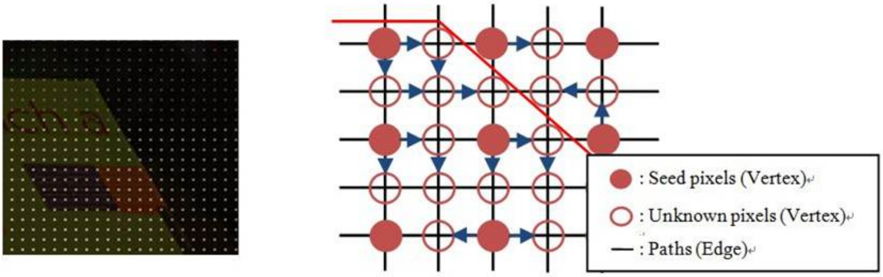
where $\boldsymbol{X_r}$ means the position in the real world coordinates of a pixel $\boldsymbol{x_r}$ in the depth map, and $d_r(\boldsymbol{x_r})$ is the return value of the corresponding depth value of $\boldsymbol{x_r}$. After finding position in the world coordinates, we then reproject the 3-D points into the color image. Equation (5) represents reporjection equation which is composed of camera parameter of the color camera and the geometric position of the depth map.

$$\mathrm{x_t = P_t X_r}\ . \tag{5}$$

where $\boldsymbol{x_t}$ is the corresponding position of $\boldsymbol{x_r}$ in the depth map. All pixels of the depth map have the position corresponded a color image through 3-D warping. However the 3-D warping technique cannot guarantee perfectly matching positions of the depth map and the color image.

3.2 Depth Up-Sampling Using Random Walk

We classify pixels as seed pixels and unknown pixels. If pixels have initial depth values, they are seed pixels and other pixels are unknown pixels. We assume that depth values of neighboring pixels which have similar color values are similar. Thus, we copy the depth values of unknown pixels from the depth values of a seed pixel which has similar color values and the low distance cost. Figure 3 shows the concept of our up-sampling method.



(a) Initial depth value

(b) Example of depth interpolation

Fig. 3. Depth map interpolation

We calculate the random walk probability of each seed pixels. Figure 4 is an example of the random walk probability of seed pixels. After calculation of the probability, unknown pixels have the probability values corresponded with each seed pixels. Thus we copy depth values of unknown pixels from that of a seed pixel which has the largest probability values.

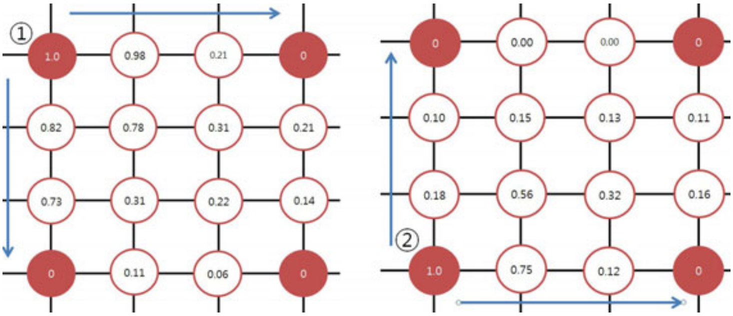


Fig. 4. Example of random walk probability

To calculate random walk probability each path between neighboring pixels has the cost. The cost between pixel *i* and pixel *j* represents

$$w_{ij} = \exp\left(-\frac{\left| z_i - z_j \right|^2}{\sigma} \right). \tag{6}$$

where $\left| z_i - z_j \right|$ represents the Euclidean color distance and σ means the variance. If unknown pixel is far from the seed pixel, random walk probability decreases as cost between two pixels. However since there are many path between the seed pixel and the unknown pixel, random walk probability depends on the path between pixels. Figure 5 shows the random walk probability corresponding paths.

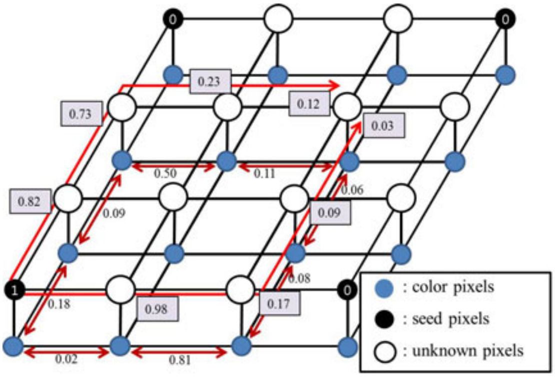


Fig. 5. Random walk probability of each path

We define the random walk probability as the largest probability among each path's probability. The largest probability means that the sum of path cost is the smallest. Thus random walk probability is

$$P(j|seed(i)) = \operatorname{argmin}\,(\sum_{i \to j} w_{ij})\,. \tag{7}$$

i is seed pixel and *j* is unknown pixel.

To find minimum cost there several methods. Graph-Cut is widely used in variety field for optimization. Graph-Cut method minimizes weight of connections between groups. However it only considers external cluster connections. It does not consider internal cluster density.

We solve this minimum cost path problem using spectral graph theory [14]. It solves the problem using simple matrix calculation. The graph is made of edges and vertices. The vertices mean the pixels and the edges mean the connectivity information. As a neighborhood system (4-neighbor or 8-neighbor), the number of edges and the shape of graph are different. We select 4-neighbor system. As shown in Figure 5, each pixel is the vertex and the connected lines are edges. All edges have cost values as defined in Eq. (6), represented as Gaussian weighting color distribution. The variance of Eq. (6) controls weighting of color and distance. Large variance increases color weighting. If variance is small, the distance weighting is larger than color weighting.

The desired random walk problem has the same solution as combinatorial Dirichlet problem [15], [16]. The Dirichlet integral is defined as

$$D[u] = \frac{1}{2}\int \left|\nabla u\right|^2 d\Omega \,. \tag{8}$$

The harmonic function satisfies the Laplace equation. Since the Laplace equation is the Euler-Lagrange equation for the Dirichlet integral, the harmonic function minimizes the Dirichlet integral [17]. Finally, the Dirichlet integral is the same and it is defined as

$$D[x] = \frac{1}{2}(Ax)^T C(Ax) = \frac{1}{2}x^T Lx = \frac{1}{2}\sum_{i,j\in E} w_{ij}(x_i - x_j)^2 \,. \tag{9}$$

Equation (7) is substituted matrix calculation $x^T Lx$ in Eq. (9). L represents the Laplacian matrix. Equation (10) represents the Laplacian matrix. w_{ij} is cost between each neighboring pixel and d_i is sum of cost between pixel i and neighboring pixels.

$$L_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -w_{ij} & \text{if } v_i \text{ and } v_j \text{ are adjancent,} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

$x^T Lx$ of Eq. (9) is

$$\begin{aligned} D[x] &= \frac{1}{2}[x_{Seed}^T \quad x_{Unknown}^T]\begin{bmatrix} L_{Seed} & B \\ B & L_{Unknown} \end{bmatrix}\begin{bmatrix} x_{Seed} \\ x_{Unknown} \end{bmatrix} \\ &= \frac{1}{2}(x_{Seed}^T L_{Seed} x_{Seed} + 2x_{Unknown}^T B^T x_{Seed} + x_{Unknown}^T L_{Unknonw} x_{Unknown}) \,. \end{aligned} \tag{11}$$

where B is a combination of the seed matrix and the unknown matrix. Since we minimize Eq. (11), we find the critical point. To find critical point, we differentiate the Eq. (11).

$$L_{Unknown} x_{Unknown} = -B^T x_{Label} \,. \tag{12}$$

Because we know B, x_{Seed}, and $L_{Unknown}$, we can find the probability $x_{Unknown}$ from Eq. (12) and fill the unknown pixel with the depth value of the seed pixel which has maximum probability. Because the Laplacian matrix is too large, solving the Eq. (12) is difficult. However since the Laplacian matrix is symmetric and sparse, it is easily solved.

4 Experimental Results

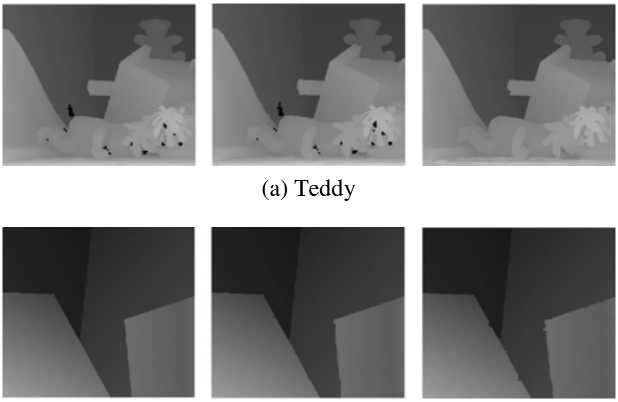
4.1 Depth Map Interpolation

To evaluate the objective performance of the proposed interpolation method, we use the data set of *Middlebury* website [18]. We apply the proposed method with the down-sampled depth maps. The down-sampled depth map consists of pixels which are on positions of multiples of up-sampling rate in the original depth map. To improve accuracy we interpolate the depth value of a block unit. Block based interpolation method is efficient because it considers only near values and positions of initial value is regular. As comparing the interpolated depth values to the original depth values, we find the error percentage which is used by *Middlebury*. In addition, we compare the error percentage with other interpolation method such as MRF refinement [9] and iterative joint bilateral filter [11]. Table 1 shows the comparison of all error percentage.

Table 1. Comparison of all error percentages

	Tsukuba			*Venus*			*Teddy*			*Cone*		
Up-sampling rate	2	4	8	2	4	8	2	4	8	2	4	8
MRF	2.51	5.12	9.68	0.57	1.24	2.69	2.78	8.33	14.5	3.55	7.52	14.4
Bilateral filter	1.16	2.56	6.95	0.25	0.42	1.19	2.43	5.95	11.5	2.39	4.76	11.0
Proposed method	0.69	1.23	2.33	0.18	0.27	0.31	2.92	3.91	5.98	3.01	3.67	5.37

In the low up-sampling rate, the error percentage is similar to the bilateral filter method. However as the up-sampling rate is higher, the proposed algorithm shows the much better performance than that of previous methods. Figure 6 shows the up-sampled depth map.



(a) Teddy

(b) Venus

Fig. 6. Double, quadruple, octuple interpolated images

4.2 Boundary Noise Remove

We obtain the depth map (176×144) using SR-4000 of the hybrid camera system. To interpolate the depth values of TOF cameras, we must consider the wrong initial values which are caused by 3-D warping error and depth map noise. When depth maps of TOF cameras are up-sampled, depth maps have noise. The depth map noise is caused by camera parameter errors, 3-D warping errors, and non-discontinuity depth value on boundary. Figure 7 shows the boundary noise of warped depth values.

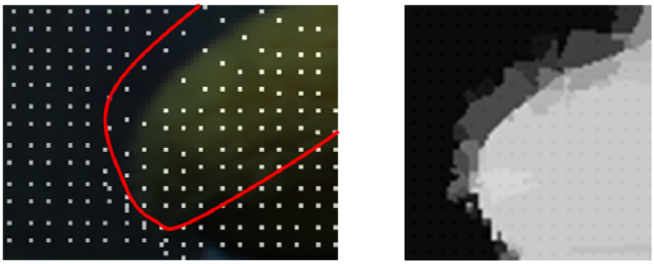
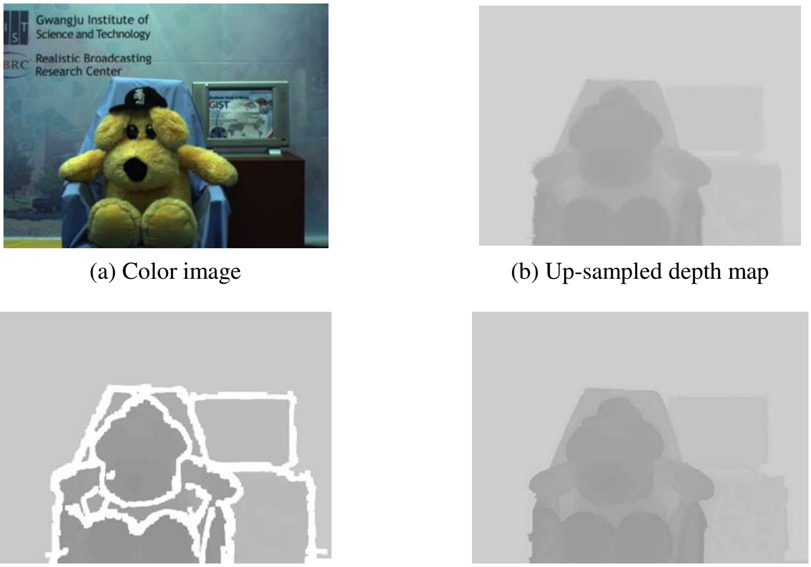


Fig. 7. Initial value noise

To overcome the problem, we redefine the depth values neighboring edges using the proposed depth hole filling method. Figure 8 shows the up-sampled depth map, color image (800×600), and boundary redefined depth map. To improve the clarity we reverse the depth values. Figure 9, 10 represent the up-sampled depth from 176×144 to 1190×950 using proposed method and 3-D rendering result.



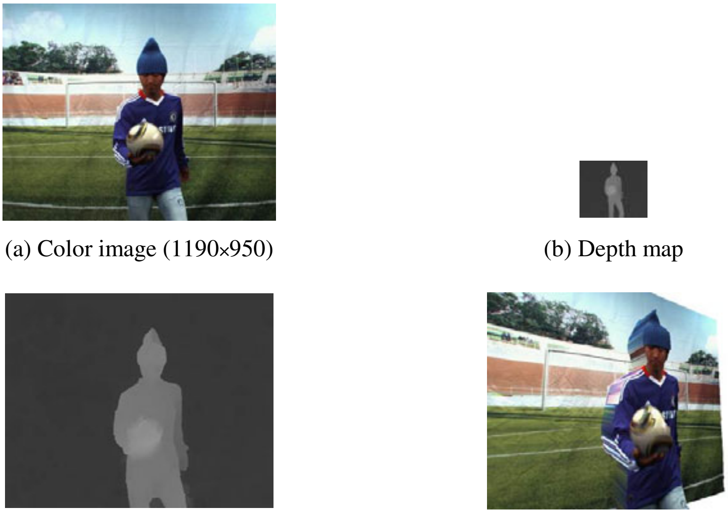
(a) Color image

(b) Up-sampled depth map

(c) Edge remove

(d) Redefined depth map

Fig. 8. Up-sampled depth map



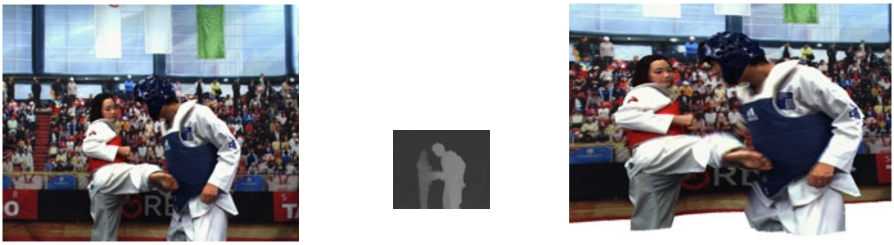
(a) Color image (1190×950)

(b) Depth map

(c) Up-sampled depth map

(d) Rendering result

Fig. 9. Up-sampled depth map and rendering result



(a) Color image (1190×950)

(b) Depth map

(c) Rendering result

Fig. 10. Rendering result using the up-sampled depth map

5 Conclusion

To render the 3-D scene, depth information is essential data. Depth maps which are captured by the depth camera cannot match color images due to resolution difference. In this paper, we propose the random walk probability model for depth up-sampling. We objectively evaluate proposed method by up-sampling *Middlebury* data sets. The proposed method enhances the accuracy of up-sampled depth maps. As the block based up-sampling rate is larger, quality variations of the proposed method is smaller than that of previous methods. Besides we enhance the depth map which is captured by TOF cameras. We interpolate the depth map using global method. And we apply the post processing to overcome problem of 3D-warping and global method. The result of proposed method shows accuracy improvement of discontinuity regions neighboring object boundary.

Acknowledgments. This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-(C1090-1111-0003)).

References

1. Kauff, P., Atzpadin, N., Fehn, C., Muller, M., Schreer, O., Smolic, A., Tanger, R.: Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability. Signal Processing: Image Communication 22(2), 217–234 (2007)
2. Nayar, S.K., Nakagawa, Y.: Shape from focus. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(8), 824–831 (1994)
3. Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. In: Proceedings of ACM SIGGRAPHS, vol. 23(3), pp. 600–608 (2004)
4. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 195–202 (2003)
5. Iddan, G., Yahav, G.: 3D imaging in the studio and elsewhere. In: Proceedings of SPIE Videometrics and Optical Methods for 3D Shape Measurements, vol. 4298, pp. 48–55 (2001)
6. Lee, E., Ho, Y.: Generation of high-quality depth maps using hybrid camera system for 3-D video. Journal of Visual Communication and Image Representation 22, 73–84 (2011)
7. Jung, J., Ho, Y.: MRF-based Depth Map Interpolation using Color Segmentation. In: Asia-Pacific Signal and Information Processing Association, pp. 19–22 (2010)
8. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C: The art of Scientific Computing. Cambridge University Press (1992)
9. Diebel, J., Thrun, S.: An application of Marko random fields to range sensing. In: Advances in Neural Information Processing Systems, vol. 18, pp. 291–298 (2005)
10. Johannes, K., Michael, C., Dani, L., Matt, U.: Joint Bilateral Upsampling. In: Proceedings of ACM SIGGRAPHS, vol. 26(3) (2007)
11. Yang, Q., Yang, R., Davis, J., Nister, D.: Spatial-depth super resolution for range images. In: International Conference on Computer Vision and Pattern Recognition (2007)
12. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1330–1334 (2000)
13. Camera Calibration Toolbox Program for Matlab provided by Caltech, `http://www.vision.caltech.edu/bouguet/calibdoc/`
14. Gardy, L.: Random Walks for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11) (2006)
15. Kakutani, S.: Markov processes and the Dirichlet problem. Proc. Jap. Acad. 21, 227–233 (1945)
16. Biggs, N.: Algebraic potential theory on graphs. Bulletin of London Mathematics Society 29, 641–682 (1997)
17. Courant, R., Hilbert, D.: Methods of Mathematical Physics, vol. 2. John Wiley and Sons (1989)
18. `http://vision.middlebury.edu/stereo/`

Evaluation of a New Coarse-to-Fine Strategy for Fast Semi-Global Stereo Matching

Simon Hermann* and Reinhard Klette

The *.enpeda..* project, Department of Computer Science
The University of Auckland, New Zealand

Abstract. The paper considers semi-global stereo matching in the context of vision-based driver assistance systems. The need for real-time performance in this field requires a design change of the originally proposed method to run on current hardware. This paper proposes such a new design; the novel strategy first generates a disparity map from half-resolution input images. The result is then used as prior to restrict the disparity search space for full-resolution computation. This approach is compared to an SGM strategy as employed currently in a state-of-the-art real-time FPGA solution. Furthermore, trinocular stereo evaluation is performed on ten real-world traffic sequences with a total of 4,000 trinocular frames. An extension to the original evaluation methodology is proposed to resolve ambiguities and to incorporate disparity density in a statistically meaningful way. Evaluation results indicate that the novel SGM method is up to 40% faster when compared to the previous strategy. It returns denser disparity maps, and is also more accurate on evaluated traffic scenes.

Keywords: Semi-global matching, driver assistance systems, coarse-to-fine stereo.

1 Introduction

Stereo correspondence analysis by *semi-global stereo matching* (SGM), as proposed by Heiko Hirschmüller [7], is a popular choice for real-time applications that require dense disparity maps at high frame rates. For example, vision-based *driver assistance systems* (DAS) favour the SGM strategy; see Rabe et al. [11]. A major constraint for real-time SGM implementation is the available memory throughput in current hardware. Because SGM integrates along multiple *1-dimensional* (1D) energy paths, a large memory block needs to be updated in off-chip memory.

Current literature on real-time SGM proposes to alter the design to the original method for ensuring high frame rates for image resolutions equivalent to the VGA norm (i.e. 640×480). For example, Hirschmüller [7] recommends to

* The first author thanks the German Academic Exchange Service (DAAD) for financial support.

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 395–406, 2011.

integrate at least along eight directions to obtain satisfactory results. But Nedevschi et al. [5] propose to integrate only along horizontal and vertical directions, leaving out diagonal energy paths. They justify their approach with the argument that objects recorded from a moving vehicle are usually aligned along the main axis, such that diagonal directions do not contribute as much to the final solution. But by omitting 50% of the accumulation procedure, the requirements on data processing are eased and real-time performance is achieved.

A research group at Daimler A.G. uses another design concept for their FPGA implementation that was proposed by Gehrig et al. [4]. They keep the recommended eight accumulation paths, but calculate a disparity image on a downscaled image pair first. The result is then scaled-up to full resolution and serves as a disparity prior. In a consecutive step they calculate a disparity map for a specified region-of-interest with SGM on full resolution images, but using only half of the disparity search space. They generate the final result by replacing disparities in the prior image with disparities from the full resolution map, if the prior suggests that a disparity lies inside the reduced search space. Otherwise the prior disparity is taken as the final result. This is based on the argument that sufficient disparity accuracy for close objects can be obtained when computing half-resolution disparity images only. But, as the re-projection error increases quadratically when disparities get smaller and boundaries of objects further away may become vague due to downscaling, it is required to calculate disparities at full resolution to minimize distance uncertainties for those objects.

The SGM design as proposed in this paper follows the Daimler approach and calculates a disparity prior on half-resolution images. However, in contrast we use the prior to actively determine the search space for the full-resolution SGM, instead of having an indication how to merge independently calculated disparity maps. Our approach therefore follows the standard coarse-to-fine concept, where results from lower-resolution images are used to initialize the same algorithm operating on the next higher resolution level. Such coarse-to-fine approaches are nowadays standard in variational motion estimation algorithms to achieve faster convergence; see, for example, the work by Brox et al. [1] or Zach et al. [18].

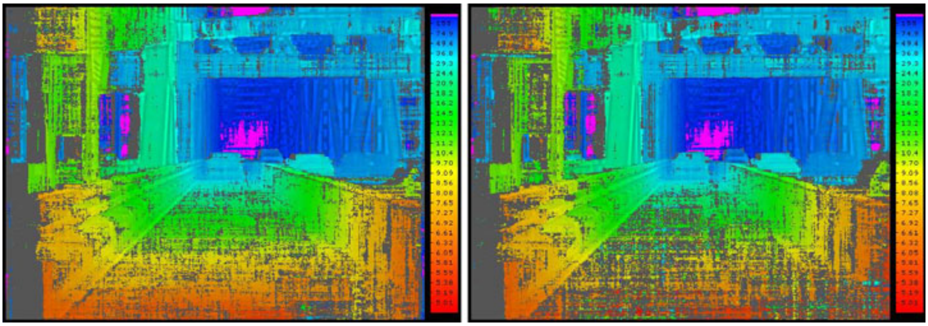


Fig. 1. Disparity results from the new SGM design (left) and the standard SGM design (right). The new design is 60% faster and is much denser especially inside the challenging road area.

To the best of our knowledge, no coarse-to-fine concept as described in the previous paragraph has been proposed so far in combination with SGM, and therefore has not been evaluated either. For the evaluation we propose an extension to an existing methodology [9] that can be used for stereo performance evaluation in the absence of ground truth and employ it on a reasonably large database of real-world traffic sequences. Of course, coarse-to-fine strategies are already employed to improve the performance of stereo matching algorithms in general. For example, a recent publication by Sizintsev and Wildes [14] employs a coarse-to-fine strategy to a block-matching algorithm. Also, in the original SGM design by Hirschmüller [7], a coarse-to-fine strategy is used, but only to support the *mutual information* (MI) cost function. The author recommends to calculate the disparities with SGM at each pyramid level from scratch. So, the prior information is just used to improve the quality of the MI cost function and not to improve the run-time performance of SGM. This defines the place where this paper is positioned, namely somewhere between the original SGM [7] and the SGM design proposed by Gehrig et al. [4]. We use design considerations from the latter work to select a method to be compared with our novel strategy, because of the shared goal to improve the run-time performance of SGM while maintaining stereo accuracy on real-world traffic scenes.

The rest of this paper is organized as follows. In Section 2, relevant details of the SGM algorithm are recalled and parameter settings of the used implementation are given. We present the design consideration as proposed by Gehrig et al., followed by our proposed coarse-to-fine approach. We also provide a discussion about run-time performance. The trinocular evaluation concept as proposed by Morales and Klette [9] is outlined in Section 3; we propose alterations and further extensions to the original method. In Section 4 we present ten real-world sequences, each of 400 trinocular frames, and outline the methodology of our experiments using trinocular evaluation. The results of this study are discussed in detail in Section 5. The paper concludes with a summary in Section 6.

2 Semi-Global Matching

We first recall the SGM algorithm and explain our alterations to the original configuration as reported in [7]. We then compare two SGM design consideration of this reference implementation. The first, called $SGM_{\mathcal{G}}$, is our implementation following the design concept as proposed by Gehrig et al. [4], and this serves as the method of comparison. The second implements our coarse-to-fine approach. We discuss the run-time and disparity analysis performance of both methods.

Cost Accumulation and Cost Function. We introduce the notation for defining the cost accumulation procedure. For a cost accumulation path $L_{\mathbf{a}}$ with direction $\mathbf{a}$, processed between image border and pixel p, we consider the segment $p_0, p_1, \ldots, p_n$ of that path, with p_0 on the image border, and $p_n = p$. The cost at pixel position p for a disparity $d \in \{0, \ldots, D\} \subset \mathbb{N}$ on the path $L_{\mathbf{a}}$ is recursively defined as follows, for $i = 1, 2, \ldots, n$:

$$L_{\mathbf{a}}(p_i, d) = C(p_i, d) + \mathcal{M}_i - \min_{\Delta} L_{\mathbf{a}}(p_{i-1}, \Delta) \tag{1}$$

with

$$\mathcal{M}_i = \min \begin{cases} L_{\mathbf{a}}(p_{i-1}, d) \\ L_{\mathbf{a}}(p_{i-1}, d-1) + c_1 \\ L_{\mathbf{a}}(p_{i-1}, d+1) + c_1 \\ \min_{\Delta} L_{\mathbf{a}}(p_{i-1}, \Delta) + c_2(p_i) \end{cases} \tag{2}$$

where $C(p, d)$ is the similarity cost of pixel p for disparity d, and c_1 and c_2 are the penalties of the smoothness term. The second penalty c_2 is individually adjusted at each pixel p_i to $c_2(p_i)$. The magnitude of the forward difference in direction $\mathbf{a}$ scales the penalty for each p_i with

$$c_2(p_i) = \frac{c_2}{|I(p_{i-1}) - I(p_i)|} \tag{3}$$

where $I(\cdot)$ refers to the intensity at a pixel. For disparities $d = 0$ and $d = D$, the terms $L_{\mathbf{a}}(p_{i-1}, d-1) + c_1$ and $L_{\mathbf{a}}(p_{i-1}, d+1) + c_1$ are removed from $\mathcal{M}_i$, respectively.

The standard SGM algorithm uses eight paths for accumulation (up, down, left, right, and the four in-between angles). To enforce uniqueness, two disparity maps are calculated to perform a left-right consistency check. A disparity passes this test if corresponding disparities do not deviate by more than one disparity level. To identify an occlusion or mismatch, a unique *invalid label* is assigned to pixels whose disparities failed this test. Disparities are calculated with sub-pixel accuracy using the equiangular interpolation method proposed by Shimizu and Okutomi [13]. The penalties are set to $c_1 = 30$ and $c_2 = 150$ for an intensity domain of $[0, 255]$. The input images are smoothed with a small 3×3 mean kernel. As similarity cost, we employ the census cost function which is based on the census transform. Several studies [8,6] found that this function is very 'descriptive' and robust, even under strong illumination variations, which is crucial for real-world applications.

The census transform [16] assigns to each pixel in the left and right image a signature vector, which is stored as a bit string (i.e. as an integer). This transformation is performed once prior to cost calculation, and signatures are stored in an integer matrix of the dimension of the image. The signature sequence is generated as follows:

$$\text{census}_{\text{sig}} = \Big[\varPsi(I_{i,j} \geq I_{i+x,j+y})\Big]_{(x,y)\in\mathcal{N}} \tag{4}$$

where $\varPsi(\cdot)$ returns 1 if true, and 0 otherwise. $\mathcal{N}$ denotes a neighbourhood (e.g. 8-neighbourhood) centred at the origin.

The census cost is the Hamming distance of two signature vectors and can be calculated very efficiently [15]. In fact, the cost of calculating the Hamming distance is proportional to the actual Hamming distance and not to the length of the signature string. This is useful in GPU implementations: calculating the cost from scratch is here cheaper than accessing the global memory [3].

Design Considerations. First we introduce some terminology. A standard SGM implementation was described in the previous subsection. We now describe the design consideration reported by Gehrig et al. [4], denoted by $SGM_{\mathcal{G}}$. Our new approach is denoted by $SGM_{\mathcal{F}}$, where subscript $\mathcal{F}$ stands for "fast".

Both programs, $SGM_{\mathcal{G}}$ and $SGM_{\mathcal{F}}$, calculate a dense disparity map applying standard SGM on half-resolution input images. The images were scaled down using a 5×5 Gauss kernel with $\sigma = 1$. The half-resolution disparity maps are scaled up; in-between pixels are linearly interpolated if both neighbours have a valid disparity assigned to them. When identifying (by the left-right consistency check) a case of occlusion or mismatch, we assign an invalid label to the corresponding 3×3 neighbourhood. This calculated half-resolution disparity map $\mathcal{P}$ serves in both methods as prior for subsequent calculations.

In case of $SGM_{\mathcal{G}}$, a second disparity map F is calculated on full-resolution input images. However, the maximum disparity D is reduced to D/2 to reduce the memory to be processed. The final disparity map R is created as follows

$$R_{i,j} = \begin{cases} \mathcal{P}_{i,j} & \text{if } \mathcal{P}_{i,j} > D/2 - 1 \\ F_{i,j} & \text{otherwise} \end{cases} \tag{5}$$

In case of $SGM_{\mathcal{F}}$, the prior $\mathcal{P}$ is used to define the search space for every individual pixel. For a valid disparity δ in $\mathcal{P}$, we process Equation (1) not for $d \in \{0, \dots, D\} \subset \mathbb{N}$ but only for $d \in \{\delta - 4, \delta - 3 \dots, \delta + 3, \delta + 4\} \subset \mathbb{N}$.

In other words we restrict the disparity search space to nine pixels around the prior. In case of disparities close to 0 or D, we do not reduce the search space but shift it accordingly. In case of invalid pixels we simply assign the default search space which would be $d \in \{0, \dots, D\} \subset \mathbb{N}$, to allow for all possible disparities.

Run-Time Performance. We analyse the approximate run-time performance on images with resolution W×H. We assume that the maximum possible disparity is D. This means that a memory block of W×H×D has to be processed, which resides in off-chip memory. Because one individual integration step consists of a constant number of operations [see Equation (1)], the run-time performance can be related to the size of the memory that needs to be processed. The advantage of this model is its independence from any hardware consideration or implementation.

The memory block used in standard SGM serves as reference to define a coefficient ϱ_X that indicates the ratio of memory needed in SGM_X. Without alterations, we have $\varrho_{\mathcal{S}} = 1$ in standard SGM.

In case of $SGM_{\mathcal{G}}$, we have to process a memory block of size W/2×H/2×D/2 for the half resolution image, and W×H×D/2 for the full resolution image. Adding those two quantities results in $\frac{5}{8}$×W×H×D, which gives a coefficient $\varrho_{\mathcal{G}} = \frac{5}{8}$. We can now measure the performance gain of $SGM_{\mathcal{G}}$ compared to standard SGM, taking into account that

$$1 - \frac{\varrho_{\mathcal{G}}}{\varrho_{\mathcal{S}}} = \frac{3}{8} = 37.5\% \tag{6}$$

In case of $SGM_{\mathcal{F}}$, the individual run-time depends on the density of the half-resolution disparity map, because the whole search space is considered at occlusions in the full-resolution run. We denote the density of this map by φ. The total memory to be processed equals

$$[W/2 \times H/2 \times D/2] + [W/2 \times H/2 \times (9\varphi + (1-\varphi)D)] \overset{!}{=} \varrho_{\mathcal{F}} \times W \times H \times D \quad (7)$$

A few algebraic operations lead to

$$\varrho_{\mathcal{F}} = \frac{9}{8} - \varphi\frac{D-9}{D} \quad (8)$$

The gain compared to $SGM_{\mathcal{G}}$ equals

$$1 - \frac{\varrho_{\mathcal{F}}}{\varrho_{\mathcal{G}}} = 1 - \frac{8}{5}\left[\frac{9}{8} - \varphi\frac{D-9}{D}\right] \quad (9)$$

We see that in case of the new design, the run-time performance can actually be worse compared to the standard SGM in cases where the prior disparity map is very sparse. However, in practice this is almost never the case; if it occurs then full-resolution SGM is well justified (i.e. the stereo data is 'challenging'). Consider on the other hand a perfectly dense prior map (i.e. $\varphi = 1$). To obtain the same run time as with $SGM_{\mathcal{G}}$, the minimum disparity range has to be at least $D = 18$. As $\varphi = 1$ is also unlikely, the performance advantage only occurs for larger values of D. For example, a common value such as $D = 128$ defines a possible run-time gain of up to 68%. We measure performance advantages in our experiments by applying Equation (9). Results below show an expected performance gain of about 40%.

3 Trinocular Stereo Evaluation

A predicted-error technique was first employed by Morales and Klette [9] for evaluating stereo analysis on long real-world stereo sequences. It requires at least stereo triples of the same scene, recorded at the same time instance by three calibrated cameras. Two of the three images (i.e reference and match image) are used to calculate a disparity map by the stereo matching algorithm of choice. Each pixel of the reference image is then projected into the position in which it would be located in the third (i.e. control) image C. This *virtual* image V is then compared to the control image C by calculating the *normalized cross-correlation* (NCC) index as follows:

$$NCC(V,C) = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \frac{[V(i,j) - \mu_V][C(i,j) - \mu_C]}{\sigma_V \sigma_C} \quad (10)$$

where μ_V and μ_C denote the means, and σ_V and σ_C the standard deviations of the control and virtual images, respectively. The domain Ω contains only non-occluded pixels (i.e. pixels which are successfully mapped from the reference image to the virtual image domain).

Generating the Virtual Image. In the original work by Morales and Klette [9] it is proposed to use a forward mapping to generate the virtual image. In other words, intensities of the reference image are mapped to positions in the control image and assigned to the closest pixel position. The problem here is that during the mapping process more than one intensity value may be mapped to the same pixel location. Discarding any of these mappings would cause a bias in the evaluation, as the final index is affected by removing potentially wrong or correct disparities. To avoid this bias we do not calculate a virtual image but rather calculate a control intensity by means of bilinear lookup from the calculated position in the control image that is compared with the intensity of the reference image.

To make this process as easy as possible and to avoid any bias from an otherwise required de-rectification step, we recorded with a horizontally aligned trinocular camera system and rectified the images with respect to the left-most camera. This way we obtain three rectified images where corresponding epipolar lines in all three images are running along the same image row. Thus, a pixel position in the control image has the same y-coordinate as the corresponding pixel in the reference image.

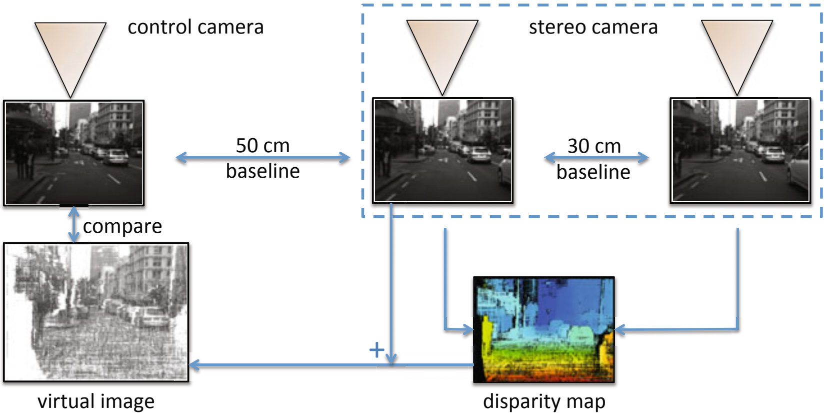


Fig. 2. Setup for the trinocular stereo experiment in this paper, showing one example frame from the experimental database

The x-coordinate is then calculated as the current location in the reference image plus an offset, which is the product of the current disparity and the ratio of the baselines from the reference image to the control and to the match image. Figure 2 shows the setup in our experiments. The stereo camera has a 30 cm baseline and a disparity map is calculated with the centre image as reference. Then the virtual image is generated and compared with the control image. The scale factor to multiply the disparities with is here $\frac{50}{30}$. Remember that in practice we warp the control image to the image plane of the reference camera as discussed before, but we will stick with the previous terminology as it makes it easier when proposing the following alteration to the original index.

Comparing Two Stereo Algorithms. The basic idea of trinocular stereo evaluation is, of course, to have a quality measure to compare the performance of different stereo algorithms in the absence of ground truth. Following the original approach, the difference of the NCC index at each frame for each stereo algorithm is evaluated. In case of only two stereo algorithms, we introduce a measure ΔNCC that calculates the signed difference of two indices. This makes it easy to compare very similar results as the sign already gives an indication which algorithm performs better.

However, there is a bias in this evaluation. The density of a disparity map is not reflected. Therefore, a sparse stereo algorithm that calculates disparities only at pixels that respond to a robust feature detector would very likely perform much better in this index than a dense algorithm which also assigns disparities in case of weak confidence. The question is how to incorporate the density in the index in a meaningful way. For that, we first introduce some further notation.

We think of images as being random variables X and Y that take intensity values as events. The NCC value can be interpreted as the correlation coefficient $\rho_{X,Y} = Cov(X,Y)/(\sigma_X \sigma_Y)$ with

$$Cov(X,Y) = E[(X - EX)(Y - EY)] \tag{11}$$

So the index reflects a mean of some distribution, and it is possible to calculate the standard deviation of it, referred to by $Cov_\sigma(X,Y)$.

We consider two disparity images D_1 and D_2 that generate two virtual images V_1 and V_2, respectively, both to be compared with a control image C. For the evaluation we consider all pixels of the domain $\Omega_1 \cup \Omega_2$. The total number of this domain is $n = |\Omega_1 \cup \Omega_2|$. We determine for each disparity image the number of invalid pixels as $k_1 = n - |\Omega_1 \setminus (\Omega_1 \cap \Omega_2)|$ and $k_2 = n - |\Omega_2 \setminus (\Omega_1 \cap \Omega_2)|$. We propose for $l = \{1,2\}$ the following index for the comparison of two stereo algorithms:

$$NCC_\sigma = \frac{1}{n}\left(\left[\sum_{(i,j)\in\Omega_1} \frac{\mathcal{K}}{\sigma_{V_l}\sigma_{C_l}}\right] + \left[\frac{k_1+k_2}{2}(NCC - Cov_\sigma)\right]\right) \tag{12}$$

where $\mathcal{K} = [V_l(i,j) - \mu_V][C_l(i,j) - \mu_C]$. We omit the arguments (V_i, C) in NCC_σ, NCC, and Cov_σ for better readability.

The index works as follows. Consider $\Omega_1 = \Omega_2$, which results in $k_1 = k_2 = 0$. In this case this index will be identical to the original NCC index as proposed in Equation (10). Now consider the symmetric case that $k_1 > k_2$ and $Cov_\sigma(V_1, C) = Cov_\sigma(V_2, C) = \upsilon$. Again, the index will be identical, because we only add terms that correspond to the pre-calculated mean. However, since we can assume that $\upsilon > 0$, we add terms such that the final index decreases. If the first term is identical for both images, the index that corresponds to the denser disparity map increases. If, on the other hand, $\upsilon_1 > \upsilon_2$, $k_1 = k_2$ and the first term is again identical in both cases, then the index that corresponds to the smaller standard deviation wins. This is reasonable, as we can assume that a smaller standard deviation refers to a 'more consistent' disparity result.

To summarize, with Equation (12) we proposed an alteration to the original evaluation index. It slightly adjusts the original index such that a higher disparity density has a positive impact on the index. We propose to use the standard deviation of the covariance for the index adjustment. This is useful because it relates to the underlying data and therefore gives also an additional quality measure (see evaluation below). But, the main motivation is that it can annihilate the benefit of a higher disparity density in case that ''additional' disparity values, which do not positively contribute to the index, increase the standard deviation and therefore have a negative affect on the final result. Thus, we regulate the NCC adjustment by two parameters, which can have a compensating or amplifying effect.

4 Evaluation Methodology and Datasets

We evaluate on ten trinocular sequences that show urban and rural environments. Each sequence consists of 400 frames. Figure 3 shows some frame samples. We refer to them by numbers only as we do not discuss them in the context of the scene they are showing, but the sample frames may help to 'read' Table 1.

Fig. 3. From left to right: Example frames of sequence, 3, 5, 9, 10, 6

We evaluated $SGM_{\mathcal{F}}$ and $SGM_{\mathcal{G}}$ on each frame of all sequences using the trinocular evaluation as proposed in Section 3. We calculated the signed difference of several values except the performance where values relating to $SGM_{\mathcal{F}}$ constitute the first summand. This list describes the results provided in Tab. 1:

- ΔNCC: difference of the original index.
- ΔNCC$_\sigma$: difference of the adjusted index.
- $\Delta\sigma$: difference of calculated Cov_σ
- Δdensity: difference of the disparity density over the whole image.
- perf. gain: the run-time gain of $SGM_{\mathcal{F}}$ compared to $SGM_{\mathcal{G}}$.

Table 1. Table of evaluation results

Seq. #	Δ NCC		Δ NCC$_\sigma$		Δ σ		Δ density		perf. gain	
1	0.73	0.33	2.13	1.29	-16.0	8.30	3.70	1.82	49.2	5.20
2	0.24	0.14	0.79	0.27	3.79	3.80	6.69	1.79	50.5	2.58
3	0.20	0.38	1.00	0.40	4.75	2.11	6.37	0.65	35.8	2.58
4	0.48	0.42	1.62	0.80	6.06	4.19	7.95	0.83	31.4	6.72
5	1.14	0.46	3.19	1.20	-6.42	8.93	4.66	0.62	44.1	4.37
6	0.25	0.40	-0.04	0.39	8.76	4.70	3.19	1.19	42.1	2.94
7	0.32	0.13	1.40	0.73	-3.63	1.97	8.25	1.73	58.9	1.64
8	0.24	0.23	1.54	1.17	-2.17	5.49	6.99	3.25	40.7	8.90
9	0.10	0.22	0.40	0.49	0.37	3.28	4.38	2.51	48.4	7.19
10	0.85	0.21	5.79	1.39	-12.7	8.53	14.8	2.01	45.6	1.60
Mean	0.46	0.27	1.78	0.81	-1.71	5.13	6.70	1.64	44.7	4.37
StdDev	0.34	0.15	1.67	0.42	8.15	2.62	3.34	0.85	7.81	2.54
Median	0.28	0.28	1.47	0.77	-0.9	4.45	6.53	1.76	44.9	3.67

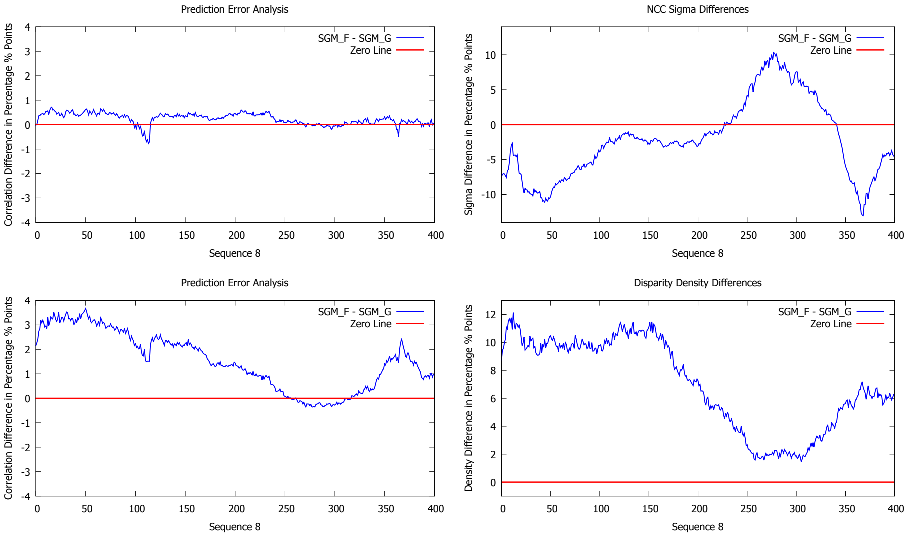


Fig. 4. Results of Sequence 8. Top: ΔNCC and $\Delta\sigma$. Bottom: ΔNCC$_\sigma$ and Δdensity.

The left entry for each item is the mean over the whole image sequence; the right entry is the standard deviation. At the bottom of the table, mean standard deviation and median are given for each item. A positive value favours $SGM_{\mathcal{F}}$ except for $\Delta\sigma$ where a negative Δ_σ defines 'better'.

Highlighted entries show relatively better performances for $SGM_{\mathcal{G}}$ (red / sequence 6), and for $SGM_{\mathcal{F}}$ (green / sequence 10). For a more detailed illustration of one sequence, see frame-by-frame results for Sequence 8 in Fig. 4; values for this sequence are close to medians and thus 'kind of representative'

Disparities of these images increase to up to 84, but we decided to run the algorithms on $D = 128$, for two reasons: First, this disparity limit is the current standard for real-time DAS stereo systems; second, the fact that this way most of the disparity map is taken from the full resolution disparity image in case of $SGM_{\mathcal{G}}$ is considered beneficial according to Gehrig et al. (page 136,[4]) who state that "Ideally, SGM would be computed everywhere at full resolution".

5 Results

Looking at performance indices ΔNCC and ΔNCC$_\sigma$ at Tab. 1 a clear tendency in favour for $SGM_{\mathcal{F}}$ is obvious. All index differences are positive with one exception in Sequence 6. However, since these values refer to percentage point differences, the performance quality of both methods is very similar. But this result comes with a mean run-time improvement of 40% over all sequences for $SGM_{\mathcal{F}}$. Along with that the new design return 5% to 6% denser disparity maps than $SGM_{\mathcal{G}}$.

To summarize, our compressed results over 4000 real-world traffic stereo frames suggest that we get slightly denser disparity maps and a positive tendency in stereo performance with a run time improvement of 40% over the method of comparison, which already has a run-time advantage of 37.5% to the standard SGM design. As we already mentioned, we defined the disparity range $D = 128$ but find actual disparities only up to 84. Therefore, the major part of the disparity map generated by $SGM_{\mathcal{G}}$ consists of the full resolution SGM. Thus, qualitative conclusion most likely hold against the standard SGM design. Different configurations and designs will be evaluated in the future, the scope of this paper only allows to introduce the new design and compare it with one design that follows a similar approach and is state-of-the-art.

We can also use the table to check the new evaluation index for consistency. See Sequence 6, where $SGM_{\mathcal{G}}$ performs best w.r.t. the new index. In this sequence we also have a low density advantage for $SGM_{\mathcal{F}}$ which is well below the mean and we have a very high Cov_{σ}. These three values are consistent with our motivation for this index. Also, Sequence 10 where $SGM_{\mathcal{F}}$ performs best has a very low Cov_{σ} and a high disparity density compared to $SGM_{\mathcal{G}}$. This also supports our argument.

For further analysis and to give an example, we picked sequence 8. for frame-by-frame analysis. We choose this sequence as it is close to the median performance (compare with final row of Tab. 1). The graphs for the first four evaluation differences of Tab. 1 can be seen in Fig. 1.

Consider the part from Frame 1-150. The old and the new index follow the same pattern, but the new index pushes $SGM_{\mathcal{F}}$ on a higher index level. Looking at $\Delta\sigma$ and $\Delta density$ we get the explanation. We have a much higher density and lower Cov_{σ} than $SGM_{\mathcal{G}}$. Here both factors work in combination. Between Frames 150 and 250 the original index stays constant and even increases a little. The new index however, slightly decreases. Looking again to the right side of the figure we see that also the density decreases and the Cov_{σ} increases. Again, this effect is visible in the new index. Between Frame 250 and 350 we have a positive impact for the method of comparison. The $\Delta\sigma$ is here in favour for $SGM_{\mathcal{G}}$. There is a slightly higher disparity density for $SGM_{\mathcal{F}}$ that has a small compensation affect. However, this again shows, that the new index works as intended and that results are conform with the expectations. We could not find an example in our results that has a contradicting tendency.

6 Conclusions

We proposed a new design for SGM that employs a coarse-to-fine strategy to reduce computational complexity. We compared this new method to a design that follows a similar approach but with a very different implementation. The common goal of both designs is to reduce the run-time of the algorithm while keeping the quality of results of the original algorithm. We evaluated both designs on 4,000 real-world traffic sequences. For the evaluation we extended an existing trinocular evaluation approach. Our experiments support that the proposed design results in a slightly higher density, has an overall tendency to more accurate results and also has an average run-time advantage of 40% over the other method. Furthermore, we evaluated a novel evaluation index and found that results are conform with out motivation for defining this index. This new index is of benefit for stereo evaluation when ground truth is missing.

References

1. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
2. .enpeda.. image sequence analysis test site, http://www.mi.auckland.ac.nz/EISATS
3. Ernst, I., Hirschmüller, H.: Mutual information based semi-global stereo matching on the GPU. In: Int. Symp. on Advances Visual Computing, pp. 228–239 (2008)
4. Gehrig, S.K., Eberli, F., Meyer, T.: A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 134–143. Springer, Heidelberg (2009)
5. Haller, I., Pantillie, C., Oniga, F., Nedevschi, S.: Real-time semi-global dense stereo solution with improved sub-pixel accuracy. In: Intelligent Vehicles Symp., pp. 369–376 (2010)
6. Hermann, S., Morales, S., Vaudrey, T., Klette, R.: Illumination Invariant Cost Functions in Semi-Global Matching. In: Koch, R., Huang, F. (eds.) ACCV Workshops 2010, Part II. LNCS, vol. 6469, pp. 245–254. Springer, Heidelberg (2011)
7. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Computer Vision Pattern Recognition, vol. 2, pp. 807–814 (2005)
8. Hirschmüller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. IEEE Trans. Pattern Analysis Machine Intelligence 31, 1582–1599 (2009)
9. Morales, S., Klette, R.: A Third Eye for Performance Evaluation in Stereo Sequence Analysis. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 1078–1086. Springer, Heidelberg (2009)
10. Ohta, Y., Kanade, T.: Stereo by two-level dynamic programming. In: Proc. of Int. Joint Conf. on Artificial Intelligence, pp. 1120–1126 (1985)
11. Rabe, C., Müller, T., Wedel, A., Franke, U.: Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 582–595. Springer, Heidelberg (2010)
12. Saito, T., Toriwaki, J.: New algorithms for n-dimensional Euclidean distance transformation. Pattern Recognition 27, 1551–1565 (1994)
13. Shimizu, M., Okutomi, M.: An analysis of subpixel estimation error on area-based image matching. Digital Signal Processing 2, 1239–1242 (2002)
14. Sizintsev, M., Wildes, R.P.: Coarse-to-fine stereo vision with accurate 3D boundaries. Image and Vision Computing 28(3), 352–366 (2010)
15. Wegener, P.: A technique for counting ones in a binary computer. Comm. ACM 3, 322 (1960)
16. Zabih, R., Woodfill, J.: Non-Parametric Local Transform for Computing Visual Correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 800, pp. 151–158. Springer, Heidelberg (1994)
17. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In: Image Vision Computing, New Zealand, pp. 1–6 (2008)
18. Zach, C., Pock, T., Bischof, H.: A Duality Based Approach for Realtime TV-L^1 Optical Flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 214–223. Springer, Heidelberg (2007)

Theoretical Analysis of Multi-view Camera Arrangement and Light-Field Super-Resolution

Ryo Nakashima[1], Keita Takahashi[2], and Takeshi Naemura[1]

[1] The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
[2] The University of Electro-Communications, 1-5-1 Chofugaoka,
Chofu-shi, Tokyo 182-8585, Japan

Abstract. We analyzed a light-field super-resolution problem in which, with a given set of multi-view images with a low resolution, the 3-D scene is reconstructed with a higher resolution using super-resolution (SR) reconstruction. The arrangement of the multi-view cameras is important because it determines the quality of the reconstruction. To simplify the analysis, we considered a situation in which a plane is located at a certain depth and a texture on that plane is super-resolved. We formulated the SR reconstruction process in the frequency domain, where the camera arrangement can be independently expressed as a matrix in the image formation model. We then evaluated the condition number of the matrix to quantify the quality of the SR reconstruction. We clarified that when the cameras are arranged in a regular grid, there exist singular depths in which the SR reconstruction becomes ill-posed. We also determined that this singularity can be avoided if the arrangement is randomly perturbed.

Keywords: multi-view cameras, super-resolution, camera arrangement, condition number.

1 Introduction

The reconstruction of a 3-D scene from multi-view images is a challenging problem and currently the focus of active research. To improve the quality of reconstruction, recent methods [3,4,9] use the framework of super-resolution (SR) reconstruction, which is a process of restoring an underlying high-resolution (HR) image from multiple low-resolution (LR) images. The quality of SR reconstruction is determined by the number, the point spread function (PSF), and the arrangement of cameras. The last factor, which has rarely been discussed, is the main focus of this paper.

The arrangement of the cameras determines disparities (pixel shifts) between the camera images given a certain depth. These disparities affect the stability of the SR reconstruction; for example, the SR reconstruction is ill-posed if all the disparities are integers. Therefore, the cameras should be arranged in such a way to avoid this ill-posed situation and improve the well-posedness of the SR reconstruction.

The purpose of this study is to analyze the relation between the arrangement of cameras and the well-posedness of an SR reconstruction. A general framework

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 407–420, 2011.

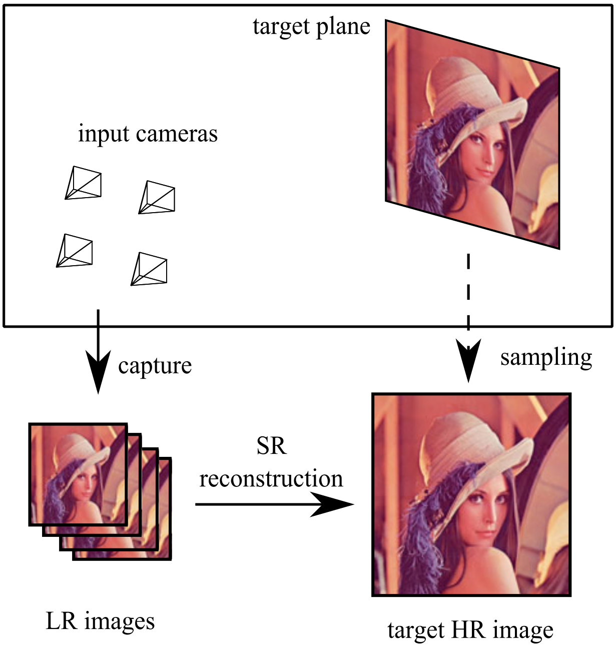


Fig. 1. Framework of LFSR

in which a high-resolution 3-D scene is reconstructed from multi-view images with a lower resolution is referred to as light-field super-resolution (LFSR). To simplify the analysis, we consider a situation in which a target plane is located at a certain depth and a texture on that plane is super-resolved, as illustrated in Fig. 1. Although we only consider a single plane at a certain depth, our analysis is applicable to general scenes with multiple objects placed at various depths because the target plane can be placed at an arbitrary depth. Our theoretical model is constructed in the frequency domain, where the camera arrangements are independently expressed as a matrix in the image formation model of the SR reconstruction. The condition number of that matrix is used to measure the well-posedness of the SR reconstruction. We determined that when the cameras are placed on a regular grid, some depths are singular, meaning that the SR reconstruction at these depths is ill-posed. Singular depths can be avoided by randomly perturbing the cameras, which is a key finding in our study.

This paper is organized as follows. Section 2 introduces related works. We formulate the SR reconstruction process in Sect. 3, followed by some descriptions of the condition number in Sect. 4. In Sect. 5, we evaluate specific camera arrangements based on our theory. Section 6 concludes the paper.

2 Related Works

SR reconstruction generally consists of two steps [5]: the registration of LR images and the reconstruction of an HR image from the registered LR images.

In this paper, we assume that registration has been done in advance and hence we focus on the reconstruction step.

The numerical performance of SR reconstruction is mainly affected by three factors: the number of LR images, subpixel shifts between LR images, and the PSF of LR images. The first and second factors are closely related; if we have a greater number of images, we are more likely to have more varied subpixel shifts, resulting in more stable SR reconstruction. However, SR reconstruction is ill-posed if all the pixel shifts are integers, no matter how many images are available.

The pixel-shifts factor has been analyzed in several other works. Robinson et al. [6] evaluated the numerical performance of SR reconstruction using the Cramér-Rao lower bound and demonstrated that reconstruction quality is maximized when the sampling points of the LR images are evenly distributed. Champagnat et al. [2] used Monte Carlo simulations to analyze the quality of SR reconstruction when fractional parts of shifts are distributed uniformly in 0–1 pixel. They found that the reconstruction quality with random pixel shifts is moderate on average and comparable to that of optimal pixel shifts. In this study, we also analyzed the pixel shifts, but they were bounded by the camera arrangement and the depth of the scene in our problem. We used Monte Carlo simulations to analyze the arrangement of the cameras because analytical optimization is difficult for our problem.

The PSF factor is also studied using the condition number, which is widely used in linear algebra to measure the well-posedness of linear equations. Baker et al. [1] analyzed box-shaped PSFs and discovered a relation between the condition number and the magnification ratio. Tanaka et al. [8] derived condition numbers for general space-invariant PSFs assuming that an infinite number of LR images are available. Inspired by these works, we also used the condition number as a measure of the well-posedness of the SR reconstruction, although we focused on the arrangement of cameras rather than the PSF.

3 Formulation of Super-Resolution Reconstruction

We formulated an SR reconstruction in the frequency domain. Our formulation is equivalent to [7], although some parameters were rearranged to fit to our problem.

3.1 Configuration

See Fig. 2 for the configuration. Let (x, y, z) be the spatial coordinate. We assume that K cameras that capture LR images are placed on the camera plane at $z = 0$. The position of the k-th camera is denoted as $(x_k, y_k, 0)$. We also assume that all the cameras have the same focal length, pixel size, and PSF. A target plane is placed at $z = z_d$ in parallel to the camera plane. The goal of the SR reconstruction is to obtain a texture on the target plane with a resolution higher than the input LR images. We assume that the magnification ratio is 2, but our analysis can easily be extended to more general cases.

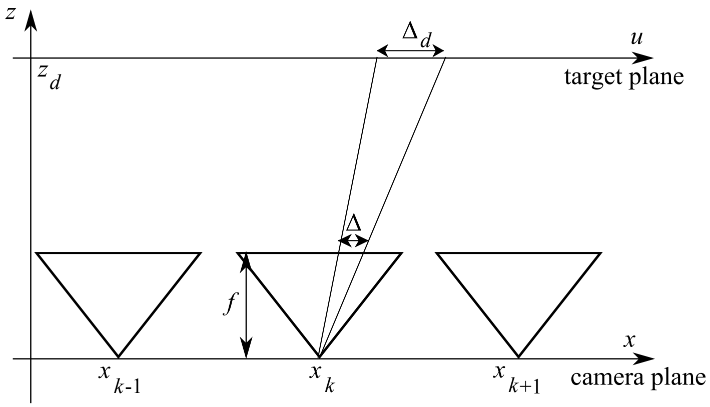


Fig. 2. Configuration analyzed in this study

Let (u, v) be the image coordinate on the target plane. The texture on it is denoted by $h(u, v)$ as a continuous 2-D signal. The HR image, which we want to synthesize by SR reconstruction, is denoted by $g_H(u, v)$. The LR image captured by the k-th camera is denoted as $g_{L,k}(u, v)$. Both $g_H(u, v)$ and $g_{L,k}(u, v)$ are the discrete signals sampled from $h(u, v)$. The pixel pitches are written as Δ and $\Delta/2$ for the LR and HR images, respectively.

3.2 Image Formation Model

The k-th LR image $g_{L,k}(u, v)$ is generated by sampling the light-rays on the focal plane. This process is equivalent to sampling the continuous texture on the target plane $z = z_d$ with intervals $\Delta_d(z_d)$, where $\Delta_d(z_d)$ is defined as

$$\Delta_d(z_d) = \frac{z_d}{f}\Delta, \tag{1}$$

where f is the focal length of the cameras, as shown in Fig. 2. Note that $\Delta_d(z_d)$ depends on the depth of the target plane z_d. To simplify the notations, we abbreviate $\Delta_d(z_d)$ as Δ_d in this section.

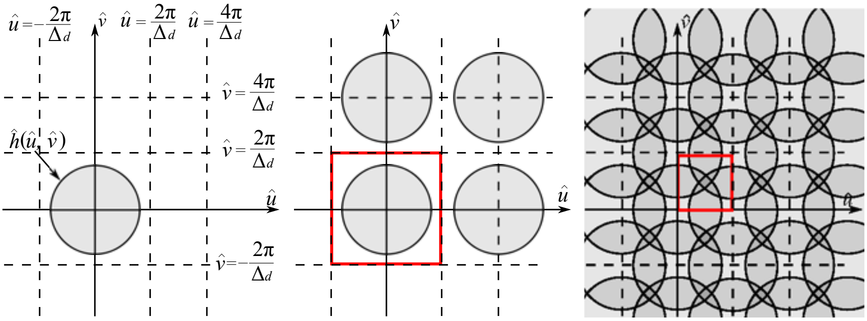
Using Δ_d, the k-th LR image $g_{L,k}(u, v)$ is defined as

$$g_{L,k}(u, v) = (h(u, v) * b_L(u, v))\, \delta_{\Delta_d}(u - x_k, v - y_k) + n_k(u, v), \tag{2}$$

where $*$ denotes convolution, $b_L(u, v)$ is a camera PSF, and $n_k(u, v)$ is the observation noise. $\delta_{\Delta'}(u, v)$ represents the sampling grid that is defined as

$$\delta_{\Delta'}(u, v) = \sum_{(m,n)\in\mathbb{Z}} \delta(u - m\Delta', v - n\Delta'), \tag{3}$$

where $\delta(u, v)$ is the Dirac delta function.


$\hat{u}=-\frac{2\pi}{\Delta_d}$
$\hat{v}$
$\hat{u}=\frac{2\pi}{\Delta_d}$
$\hat{u}=\frac{4\pi}{\Delta_d}$
$\hat{v}=\frac{4\pi}{\Delta_d}$
$\hat{h}(\hat{u},\hat{v})$
$\hat{v}=\frac{2\pi}{\Delta_d}$
$\hat{u}$
$\hat{v}=-\frac{2\pi}{\Delta_d}$
$\hat{v}$
$\hat{u}$
$\hat{v}$
$\hat{u}$

(a) Continuous image (b) HR image (c) LR image

Fig. 3. Frequency spectra of continuous, HR, and LR images

Since the resolution of the HR image is double that of the LR images, the sampling interval of the HR image is $\Delta_d/2$. Therefore, the HR image $g_H(u,v)$, whose origin is set to $(u,v)=(0,0)$, is defined as

$$g_H(u,v) = (h(u,v) * b_H(u,v))\,\delta_{\Delta_d/2}(u,v)\,, \tag{4}$$

where $b_H(u,v)$ denotes a PSF of the HR image.

3.3 Super-Resolution in the Frequency Domain

Assume that the underlying continuous image $h(u,v)$ is band-limited within $(-2\pi/\Delta_d, 2\pi/\Delta_d)$. In other words, the sampling interval of the HR image satisfies the Nyquist condition. This situation is illustrated in Fig. 3(a). The circular region in the figure represents the spectral support of the underlying continuous image $\hat{h}(\hat{u},\hat{v})$, where $\hat{}$ denotes the frequency-domain representation of the corresponding symbol.

Here, we want to obtain the Fourier transform of (4) and (2). First, we obtain the Fourier transform of (3) as

$$\hat{\delta}_{\Delta'}(\hat{u},\hat{v}) = \frac{4\pi^2}{\Delta'} \sum_{\{m,n\}\in\mathbb{Z}} \delta\left(\hat{u}-\frac{2m\pi}{\Delta'}, \hat{v}-\frac{2n\pi}{\Delta'}\right). \tag{5}$$

This equation represents a spectral replication in the frequency domain caused by discretization. For the case of the HR image, where $\Delta' = \Delta_d/2$, the repeating cycle is $(4\pi/\Delta_d, 4\pi/\Delta_d)$, as shown in Fig. 3(b). Since the original signal $\hat{h}(\hat{u},\hat{v})$ is band-limited within $(-2\pi/\Delta_d, 2\pi/\Delta_d)$, no overlapping occurs in the frequency domain. Consequently, for the frequency $\hat{u},\hat{v} \in (-2\pi/\Delta_d, 2\pi/\Delta_d)$, $\hat{g}_H(\hat{u},\hat{v})$ can be written as

$$\hat{g}_H(\hat{u},\hat{v}) = \frac{16\pi^2}{\Delta_d^2}\hat{h}(\hat{u},\hat{v})\hat{b}_H(\hat{u},\hat{v}). \tag{6}$$

Meanwhile, for the case of the LR image, where $\Delta' = \Delta_d$, the repeating cycle is $(2\pi/\Delta_d, 2\pi/\Delta_d)$. As shown in Fig. 3(c), four spectral components overlap in the range $\hat{u},\hat{v} \in (0, 2\pi/\Delta_d)$. Therefore, for this range, $\hat{g}_{L,k}(\hat{u},\hat{v})$ is described as

$$\hat{g}_k(\hat{u},\hat{v}) = \frac{4\pi^2}{\varDelta_d^2}\hat{h}\left(\hat{u},\hat{v}\right)\hat{b}_L(\hat{u},\hat{v}) * \sum_{m,n\in\{0,1\}}\delta\left(\hat{u}-\frac{2m\pi}{\varDelta_d},\hat{v}-\frac{2n\pi}{\varDelta_d}\right)e^{-j(x_k\hat{u}+y_k\hat{v})}$$
$$+\hat{n}(\hat{u},\hat{v}). \tag{7}$$

Using (6), we obtain

$$\hat{g}_k(\hat{u},\hat{v}) = \frac{1}{4}\ \hat{g}_H\left(\hat{u},\hat{v}\right)\frac{\hat{b}_L(\hat{u},\hat{v})}{\hat{b}_H(\hat{u},\hat{v})} * \sum_{m,n\in\{0,1\}}\delta\left(\hat{u}-\frac{2m\pi}{\varDelta_d},\hat{v}-\frac{2n\pi}{\varDelta_d}\right)e^{-j(x_k\hat{u}+y_k\hat{v})}$$
$$+\ \hat{n}(\hat{u},\hat{v}). \tag{8}$$

Equation (8) can be rearranged into a linear equation:

$$\hat{\mathbf{g}}_L = \hat{\mathbf{W}}\,\hat{\mathbf{g}}_H + \hat{\mathbf{n}}, \tag{9}$$

where $\hat{\mathbf{g}}_L$ represents the spectra of all LR images and $\hat{\mathbf{g}}_H$ represents the four overlapping components of the HR image:

$$\hat{\mathbf{g}}_L = \begin{pmatrix}\hat{g}_{L,1}(\hat{u},\hat{v})\\ \hat{g}_{L,2}(\hat{u},\hat{v})\\ \vdots\\ \hat{g}_{L,K}(\hat{u},\hat{v})\end{pmatrix}, \hat{\mathbf{g}}_H = \begin{pmatrix}\hat{g}_H(\hat{u}, \hat{v})\\ \hat{g}_H(\hat{u}-\frac{2\pi}{\varDelta_d}, \hat{v})\\ \hat{g}_H(\hat{u}, \hat{v}-\frac{2\pi}{\varDelta_d})\\ \hat{g}_H(\hat{u}-\frac{2\pi}{\varDelta_d}, \hat{v}-\frac{2\pi}{\varDelta_d})\end{pmatrix}. \tag{10}$$

$\hat{\mathbf{W}}$ represents the image formation model, expressed as

$$\hat{\mathbf{W}} = \hat{\mathbf{M}}\hat{\mathbf{B}}\,, \tag{11}$$

$$\text{where } \hat{\mathbf{M}} = \begin{pmatrix}1 & e^{-2j\pi x_1/\varDelta_d} & e^{-2j\pi y_1/\varDelta_d} & e^{-2j\pi(x_1+y_1)/\varDelta_d}\\ 1 & e^{-2j\pi x_2/\varDelta_d} & e^{-2j\pi y_2/\varDelta_d} & e^{-2j\pi(x_2+y_2)/\varDelta_d}\\ \vdots & \vdots & \vdots & \vdots\\ 1 & e^{-2j\pi x_K/\varDelta_d} & e^{-2j\pi y_K/\varDelta_d} & e^{-2j\pi(x_K+y_K)/\varDelta_d}\end{pmatrix}, \tag{12}$$

$$\hat{\mathbf{B}} = \text{diag}\begin{pmatrix}\hat{b}_L(\hat{u}, \hat{v})/\hat{b}_H(\hat{u}, \hat{v})\\ \hat{b}_L(\hat{u}-\frac{2\pi}{\varDelta_d}, \hat{v})/\hat{b}_H(\hat{u}-\frac{2\pi}{\varDelta_d}, \hat{v})\\ \hat{b}_L(\hat{u}, \hat{v}-\frac{2\pi}{\varDelta_d})/\hat{b}_H(\hat{u}, \hat{v}-\frac{2\pi}{\varDelta_d})\\ \hat{b}_L(\hat{u}-\frac{2\pi}{\varDelta_d}, \hat{v}-\frac{2\pi}{\varDelta_d})/\hat{b}_H(\hat{u}-\frac{2\pi}{\varDelta_d}, \hat{v}-\frac{2\pi}{\varDelta_d})\end{pmatrix}. \tag{13}$$

$\hat{\mathbf{M}}$ is a $K \times 4$ matrix and represents the camera arrangement. The k-th row of $\hat{\mathbf{M}}$ corresponds to the position of the k-th camera. We refer to this matrix as a camera arrangement matrix. $\hat{\mathbf{B}}$ is a 4×4 matrix and represents the PSF. We call this matrix a PSF matrix. $\hat{\mathbf{n}}$ represents the observation noises.

$$\hat{\mathbf{n}} = \begin{pmatrix}\hat{n}_1(\hat{u},\hat{v})\\ \hat{n}_2(\hat{u},\hat{v})\\ \vdots\\ \hat{n}_K(\hat{u},\hat{v})\end{pmatrix}. \tag{14}$$

The SR reconstruction is formulated as the problem of estimating $\hat{\mathbf{g}}_H$ given $\hat{\mathbf{W}}$ and $\hat{\mathbf{g}}_L$ in (9), where $\hat{\mathbf{W}}$ determines the well-posedness.

4 Condition Number of the SR Reconstruction

Given a coefficient matrix of a linear equation, the condition number of the matrix determines the stability of the solution. When the condition number is low, the linear equation is well-posed and is robust to noises. In contrast, if the number is high, the system is ill-posed and is sensitive to noises. The linear system is singular when the condition number is infinite.

The condition number of $\hat{\mathbf{W}}$ is defined as

$$\mathrm{cond}(\hat{\mathbf{W}}) = \parallel \hat{\mathbf{W}} \parallel \cdot \parallel \hat{\mathbf{W}}^{+} \parallel = \sqrt{\frac{\lambda_{\max}(\hat{\mathbf{W}}^{*}\hat{\mathbf{W}})}{\lambda_{\min}(\hat{\mathbf{W}}^{*}\hat{\mathbf{W}})}}, \tag{15}$$

where $\parallel \cdot \parallel$ denotes operator norm, * denotes conjugate transpose, $^{+}$ denotes Moore-Penrose pseudoinverse, and $\lambda_{\max}(\hat{\mathbf{W}}^{*}\hat{\mathbf{W}})$ and $\lambda_{\min}(\hat{\mathbf{W}}^{*}\hat{\mathbf{W}})$ are the maximum and minimum eigenvalues of $\hat{\mathbf{W}}^{*}\hat{\mathbf{W}}$, respectively.

The condition number gives the upper bound of the relative errors as

$$\frac{\parallel \hat{\mathbf{e}} \parallel_2}{\parallel \hat{\mathbf{g}}_H \parallel_2} \leq \mathrm{cond}(\hat{\mathbf{W}}) \frac{\parallel \hat{\mathbf{n}} \parallel_2}{\parallel \hat{\mathbf{g}}_L \parallel_2}, \tag{16}$$

where $\hat{\mathbf{e}}$ is the estimation error of $\hat{\mathbf{g}}_H$. This equation shows that the condition number can be used to estimate the reconstruction quality.

A key feature of our formulation is that $\hat{\mathbf{W}}$ is expressed as the product of $\hat{\mathbf{M}}$ and $\hat{\mathbf{B}}$, as shown in (11). This enables us to evaluate the camera arrangement by using the condition number of $\hat{\mathbf{M}}$ separately from the PSFs represented by $\hat{\mathbf{B}}$. The condition number of $\hat{\mathbf{W}}$ is upper-bounded by the condition numbers of $\hat{\mathbf{M}}$ and $\hat{\mathbf{B}}$ as

$$\begin{aligned}
\mathrm{cond}(\hat{\mathbf{W}}) &= \parallel \hat{\mathbf{M}}\hat{\mathbf{B}} \parallel \cdot \parallel (\hat{\mathbf{M}}\hat{\mathbf{B}})^{+} \parallel \\
&\leq \left(\parallel \hat{\mathbf{M}} \parallel \cdot \parallel \hat{\mathbf{B}} \parallel\right) \left(\parallel \hat{\mathbf{M}}^{+} \parallel \cdot \parallel \hat{\mathbf{B}}^{-1} \parallel\right) \\
&= \left(\parallel \hat{\mathbf{M}} \parallel \cdot \parallel \hat{\mathbf{M}}^{+} \parallel\right) \left(\parallel \hat{\mathbf{B}} \parallel \cdot \parallel \hat{\mathbf{B}}^{-1} \parallel\right) \\
&= \mathrm{cond}(\hat{\mathbf{M}}) \cdot \mathrm{cond}(\hat{\mathbf{B}})
\end{aligned} \tag{17}$$

using sub-multiplicativity of the operator norm. We also use the inverse condition number for convenience.

5 Analyses of Camera Arrangements

In this section, we analyze some specific camera arrangements using the condition number. In subsection 5.1, we analyze regular grid arrangements and show that the condition number becomes infinite at periodic depths. In subsection 5.2, we analyze grid-and-perturbation arrangements using Monte Carlo simulation.

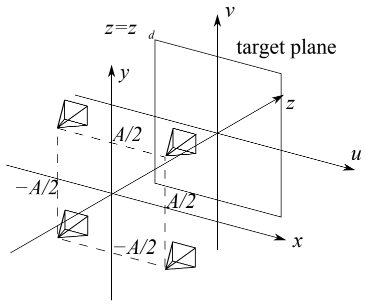


Fig. 4. A grid arrangement of 4 cameras

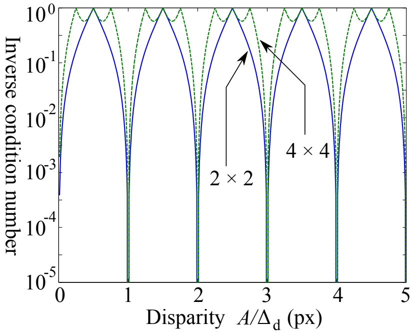


Fig. 5. Inverse condition number of a grid arrangement

5.1 Analysis on Regular Grid Arrangement

Assume that four cameras are placed on a 2×2 regular grid. Let the camera positions be $(\pm A/2, \pm A/2, 0)$, where the distance between the cameras is A, as illustrated in Fig. 4.

Using (12), the camera arrangement matrix $\hat{\mathbf{M}}$ is written as

$$\hat{\mathbf{M}} = \begin{pmatrix} 1 & e^{-j\pi A/\Delta_d} & e^{-j\pi A/\Delta_d} & e^{-2j\pi A/\Delta_d} \\ 1 & e^{-j\pi A/\Delta_d} & e^{j\pi A/\Delta_d} & 1 \\ 1 & e^{j\pi A/\Delta_d} & e^{-j\pi A/\Delta_d} & 1 \\ 1 & e^{j\pi A/\Delta_d} & e^{j\pi A/\Delta_d} & e^{2j\pi A/\Delta_d} \end{pmatrix}, \tag{18}$$

whose condition number (see appendix for derivation) is

$$\mathrm{cond}(\hat{\mathbf{M}}) = \frac{1 + \left|\cos \pi \frac{A}{\Delta_d(z_d)}\right|}{1 - \left|\cos \pi \frac{A}{\Delta_d(z_d)}\right|}. \tag{19}$$

We also analyzed a case in which 16 cameras were arranged on a 4×4 regular grid. The condition number of $\hat{\mathbf{M}}$ (see appendix for derivation) is

$$\mathrm{cond}(\hat{\mathbf{M}}) = \frac{1 + \left|\cos \frac{2\pi A}{\Delta_d(z_d)} \cos \frac{\pi A}{\Delta_d(z_d)}\right|}{1 - \left|\cos \frac{2\pi A}{\Delta_d(z_d)} \cos \frac{\pi A}{\Delta_d(z_d)}\right|}. \tag{20}$$

Figure 5 shows the inverse condition number of the camera arrangement matrix for the regular grid arrangement of 4 cameras and 16 cameras. The horizontal axis represents the value of $A/\Delta_d(z_d)$. Note that $A/\Delta_d(z_d)$ is inversely proportional to the depth of the target plane z_d and corresponds to the disparity between the input LR images.

As shown in the figure, the inverse condition number takes zero at periodic depths where $A/\Delta_d(z_d)$ is an integer. When the target plane is located at these

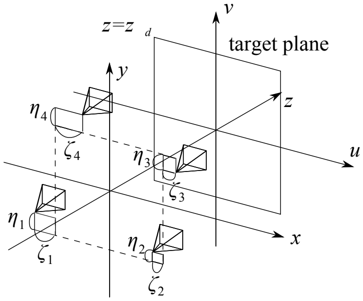


Fig. 6. An example of grid-and-perturbation arrangements of 4 cameras

depths, the sampling points of all input cameras coincide with each other. We refer to these depths as singular depths. It should be noted that the singular depths exist regardless of the number of cameras as long as they are arranged in regular grids.

It is obvious that the inverse condition number takes the maximum value at periodic depths, where the disparity is a half-integer in the case of the 2 × 2 grid and a quarter-integer in the case of the 4 × 4 grid. These depths, where the SR reconstruction is the most stable, are referred to as the best depths.

To summarize, when the cameras are arranged in a regular grid, SR reconstruction becomes ill-posed at some depths yet well-posed at other depths. This situation is undesirable in terms of reconstructing an entire 3-D scene.

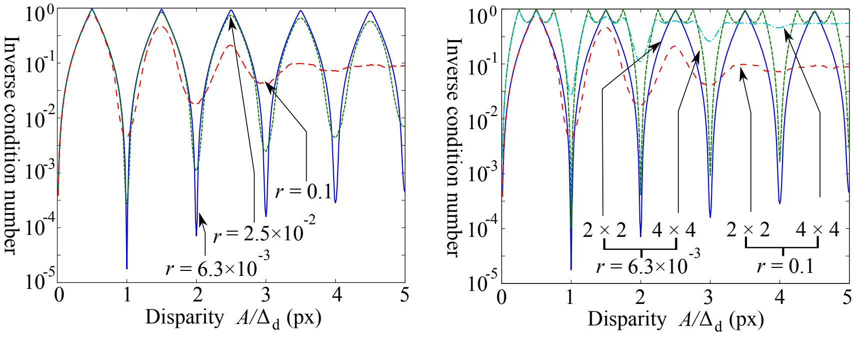
5.2 Analysis on Grid-and-Perturbation Arrangement

The periodic structure of the condition number along A/Δ_d comes from the regularity of the camera arrangement. Thereby, randomizing the camera arrangement should decrease the periodicity and might be helpful to avoid the singular depth problem. In this subsection, we analyze a case where the camera arrangement is randomly perturbed from the regular grid.

For this case, analytical derivation of the condition number is difficult, so we used Monte Carlo simulations. We randomly generated many camera arrangements and numerically computed the condition numbers of the camera arrangement matrices $\hat{\mathbf{M}}$.

Monte Carlo Simulations. The number of cameras was set to either 4 or 16. The cameras were shifted from the 2 × 2 or 4 × 4 regular grid arrangements, as shown in Fig. 6. The shift of the k-th camera is denoted as $(\zeta_k, \eta_k, 0)$, where ζ_k and η_k were independently sampled from the uniform distribution over $(-rA, rA)$. $r > 0$ is the parameter that defines the range of the distribution and was used to control the randomness of the camera arrangement.

We exponentially varied r from 10^{-10} to 1 and generated 1000 shifts for each r value. We then numerically computed the condition number of each arrangement for the range of $0 < A/\Delta_d(z_d) \leq 10$.



(a) Varying r (b) Varying the number of cameras

Fig. 7. Inverse condition number of a grid-and-perturbation arrangement

Relation between Depth and Condition Number. For each r value, we computed the geometric average of the inverse condition numbers over 1000 arrangements. The results with $r = 6.3 \times 10^{-3}$, 2.5×10^{-2}, and 0.1 are shown in Fig. 7(a). Note that the horizontal axis is $A/\Delta_d(z_d)$. When we set $r > 0$, the inverse condition number became higher than zero for the depths where they were singular with the original regular grid arrangement. As r increased, the inverse condition number also increased at these depths. For instance, when $A/\Delta_d(z_d) = 2$, the inverse condition numbers for $r = 6.3 \times 10^{-3}$, 2.5×10^{-2}, and 0.1 were $10^{-4.5}$, $10^{-2.5}$, and $10^{-1.5}$, respectively. Meanwhile, as r increased, the inverse condition number decreased at best depths. For instance, when $A/\Delta_d(z_d) = 3.5$, the inverse condition numbers for $r = 6.3 \times 10^{-3}$, 2.5×10^{-2}, and 0.1 were 1, $10^{-0.5}$, and 10^{-1}, respectively. This tendency indicates that there is a trade-off between the improvement at the singular depths and the decline at the best depths. Randomizing the arrangement is likely to flatten the performance of the inverse condition number over the depths. This trade-off is discussed in more detail in the next subsection.

We also analyzed the relation between the number of cameras and the inverse condition number. Figure 7(b) shows the results when the number of cameras was 4 or 16 and $r = 6.3 \times 10^{-3}$ or 0.1. The grid-and-perturbation arrangement was effective for both 4 and 16 cameras. As a whole, the inverse condition numbers with 16 cameras were larger than those with 4 cameras.

Relation between Randomness and Overall Image Quality. As mentioned above, randomizing the camera arrangements raised the inverse condition numbers at singular depths, but lowered them at best depths. Therefore, we introduced a new measure, referred to as overall image quality, which is the geometric average of the inverse condition numbers over the range of the entire 3-D scene. The range was set to $0 < A/\Delta_d(z_d) \leq 5$ in this experiment. Note that the overall image quality is zero for regular grid arrangements, since there are singular depths.

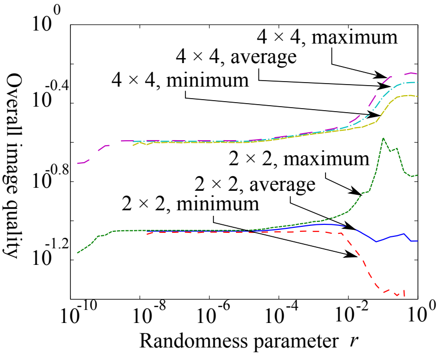


Fig. 8. Relation between randomness parameter r and overall image quality

We tested 1000 random arrangements for each r. Figure 8 shows the relation between the randomness parameter r and the overall image quality. The maximum, average, and minimum values of the overall image quality are plotted in this figure. Note that the vertical axis is logarithmic.

The average and minimum values are not plotted on the left side of the graph because their values were zero due to the machine precision. This indicates that when r is very small and the arrangement is very close to the regular grid arrangement, some depths become nearly singular. Therefore, r should not be very small.

When the number of the cameras was four, the overall image quality gradually increased when $r < 10^{-3}$ but gradually decreased when $r > 10^{-3}$. Improvements around the singular depth and degradation around the best depths seem to have balanced around $r = 10^{-3}$. When the number of the cameras was 16, the overall image quality monotonically increased in the range of $10^{-10} \leq r \leq 1$. This result indicates that randomizing the camera arrangement is more effective when more cameras are used.

The difference between the maximum and minimum overall image quality became bigger as r increased. This tendency indicates that a very large r should be avoided to control the overall image quality. It should also be noted that the worst overall image quality is still more than zero when $r > 10^{-8}$. Therefore, we can expect the overall image quality to improve by randomizing the camera arrangement even in a worst-case scenario.

6 Conclusion

In this paper, we considered the arrangement of multi-view cameras for light-field super-resolution. We formulated an SR reconstruction in the frequency domain and derived the relation between the camera arrangement and the stability of the SR reconstruction using the condition number. Based on this relation, we showed that the singular depths, where the reconstruction becomes ill-posed, periodically appear in the case of regular grid arrangements. We also revealed that randomizing the camera arrangement can prevent the singular depths and improve the stability of the SR reconstruction for the entire 3-D scene. Our future work is to verify the correctness of our theoretical analysis by experiments. We also plan to analyze more general camera arrangements.

Acknowledgments. This research is supported by the Strategic Information and Communication R&D Promotion Programs (SCOPE) of the Ministry of Internal Affairs and Communications, Japan.

References

1. Baker, S., Kanade, T.: Limits on super-resolution and how to break them. IEEE Transactions on Pattern Analysis Machine Intelligence 24(9), 1167–1183 (2002)
2. Champagnat, F., Besnerais, G.L., Kulcsár, C.: Statistical performance modeling for superresolution: a discrete data-continuous reconstruction framework. Journal of the Optical Society of America A 26(7), 1730–1746 (2009)
3. Fukushima, N., Ishibashi, Y.: Free viewpoint image generation with super resolution. In: Picture Coding Symposium, pp. 1–4 (2010)
4. Mudenagudi, U., Gupta, A., Goel, L., Kushal, A., Kalra, P., Banerjee, S.: Super Resolution Of Images of 3D Scenecs. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 85–95. Springer, Heidelberg (2007)
5. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: a technical overview. IEEE Signal Processing Magazine 20(3), 21–36 (2003)
6. Robinson, D., Milanfar, P.: Statistical performance analysis of super-resolution. IEEE Transactions on Image Processing 15(6), 1413–1428 (2006)
7. Takahashi, K., Naemura, T., Tanaka, M.: Rate-distortion analysis of super-resolution image/video decoding. In: International Conference on Image Processing (2011)
8. Tanaka, M., Okutomi, M.: Theoretical analysis on reconstruction-based super-resolution for an arbitrary PSF. In: IEEE Computer Vision and Pattern Recognition, vol. 2, pp. 947–954 (2005)
9. Tung, T., Nobuhara, S., Matsuyama, T.: Simultaneous super-resolution and 3d video using graph-cuts. In: IEEE Computer Vision and Pattern Recognition, pp. 1–8 (2008)

Appendix: The Condition Number of Regular Grid Arrangements

See Fig. 9 for the configuration.

A. 2 × 2 Cameras

Here, we derive (19). First, we compute $\hat{\mathbf{M}}^*\hat{\mathbf{M}}$, which is written as

$$\hat{\mathbf{M}}^*\hat{\mathbf{M}} = 4 \begin{pmatrix} 1 & \alpha & \alpha & \alpha^2 \\ \alpha & 1 & \alpha^2 & \alpha \\ \alpha & \alpha^2 & 1 & \alpha \\ \alpha^2 & \alpha & \alpha & 1 \end{pmatrix}, \tag{21}$$

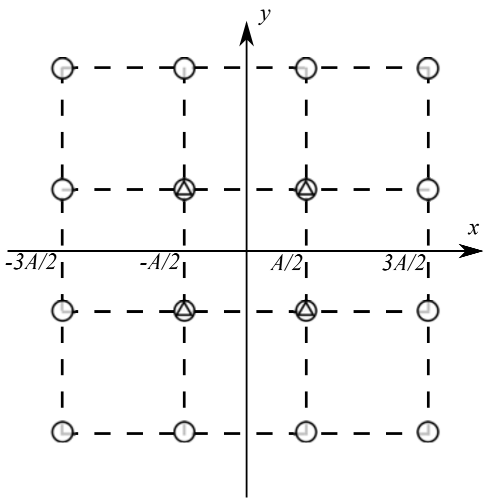


Fig. 9. Configuration. The triangles and circles represents the positions of the cameras in 2 × 2 and 4 × 4 regular grid arrangements, respectively.

where

$$\alpha = \frac{e^{j\pi A/\Delta_d} + e^{-j\pi A/\Delta_d}}{2} \tag{22}$$

$$= \cos\frac{\pi A}{\Delta_d}. \tag{23}$$

By analytically solving the eigenequation of $\hat{\mathbf{M}}^*\hat{\mathbf{M}}$, we obtain

$$\lambda = 4(1 \pm |\alpha|)^2, 4(1 - |\alpha|^2). \tag{24}$$

Note that $\lambda = 4(1 - |\alpha|^2)$ is a double root.

Since $0 \leq |\alpha| \leq 1$, these eigenvalues satisfy the relation

$$4(1 - |\alpha|)^2 \leq 4(1 - |\alpha|^2) \leq 4(1 + |\alpha|)^2. \tag{25}$$

The left equality holds when $|\alpha| = 0, 1$, and the right equality hold when $|\alpha| = 0$. Using (25), the maximum and minimum eigenvalues are

$$\lambda_{\max}(\hat{\mathbf{M}}^*\hat{\mathbf{M}}) = 4(1 + |\alpha|)^2, \tag{26}$$

$$\lambda_{\min}(\hat{\mathbf{M}}^*\hat{\mathbf{M}}) = 4(1 - |\alpha|)^2. \tag{27}$$

By substituting (23), (26), (27) into (15), we obtain (19).

B. 4 × 4 Cameras

We derive (20). Similar to the previous derivation, we compute eigenvalues of $\hat{\mathbf{M}}^*\hat{\mathbf{M}}$. In this case, $\hat{\mathbf{M}}^*\hat{\mathbf{M}}$ becomes

$$\hat{\mathbf{M}}^*\hat{\mathbf{M}} = 16 \begin{pmatrix} 1 & \alpha & \alpha & \alpha^2 \\ \alpha & 1 & \alpha^2 & \alpha \\ \alpha & \alpha^2 & 1 & \alpha \\ \alpha^2 & \alpha & \alpha & 1 \end{pmatrix}, \tag{28}$$

where

$$\begin{aligned} \alpha &= \frac{e^{3j\pi A/\Delta_d} + e^{j\pi A/\Delta_d} + e^{-j\pi A/\Delta_d} + e^{-3j\pi A/\Delta_d}}{4} \\ &= \cos\frac{2\pi A}{\Delta_d}\cos\frac{\pi A}{\Delta_d}. \end{aligned} \tag{29}$$

The form of $\hat{\mathbf{M}}^*\hat{\mathbf{M}}$ is the same as (21) except for the value of α and multiplication by a constant value, Therefore, the maximum and minimum eigenvalues are similary computed as

$$\lambda_{\max}(\hat{\mathbf{M}}^*\hat{\mathbf{M}}) = 16(1 + |\alpha|)^2, \tag{30}$$

$$\lambda_{\min}(\hat{\mathbf{M}}^*\hat{\mathbf{M}}) = 16(1 - |\alpha|)^2. \tag{31}$$

By substituting (29), (30), (31) into (15), we obtain (20).

Author Index