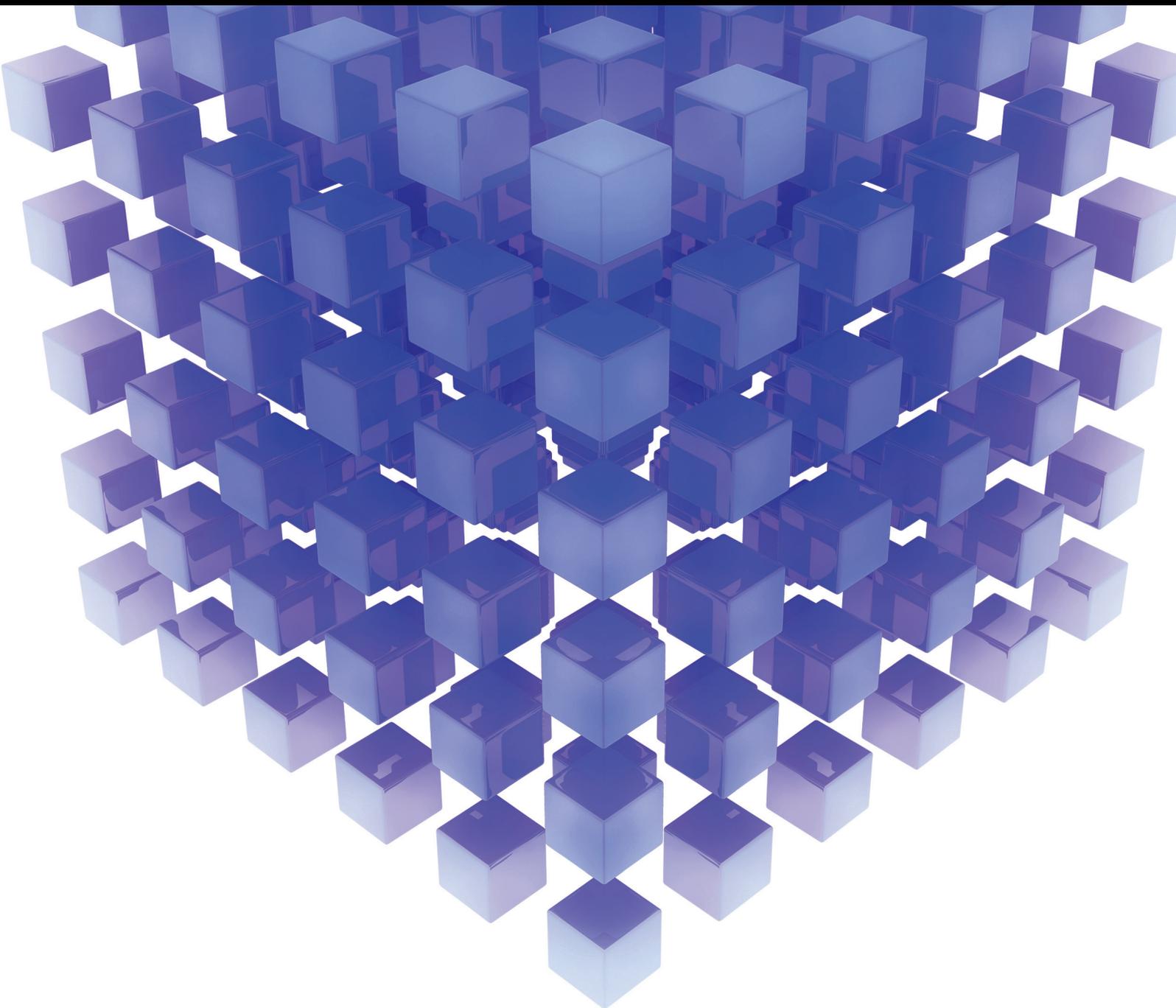


MATHEMATICAL PROBLEMS IN ENGINEERING

MATHEMATICAL METHODS APPLIED TO DIGITAL IMAGE PROCESSING

GUEST EDITORS: YI-HUNG LIU, CHUNG-HAO CHEN, AND PAUL C.-P. CHAO





Mathematical Methods Applied to Digital Image Processing

Mathematical Problems in Engineering

Mathematical Methods Applied to Digital Image Processing

Guest Editors: Yi-Hung Liu, Chung-Hao Chen,
and Paul C.-P. Chao



Copyright © 2014 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in “Mathematical Problems in Engineering.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

- M. Abd El Aziz, Egypt
E. M. Abdel-Rahman, Canada
R. K. Abu Al-Rub, USA
Sarp Adali, South Africa
Salvatore Alfonzetti, Italy
Igor Andrianov, Germany
Sebastian Anita, Romania
W. Assawinchaichote, Thailand
Erwei Bai, USA
Ezzat G. Bakhoum, USA
José M. Balthazar, Brazil
R. K. Bera, India
C. Bérenguer, France
Jonathan N. Blakely, USA
Stefano Boccaletti, Spain
Stephane P.A. Bordas, USA
Daniela Boso, Italy
M. Boutayeb, France
Michael J. Brennan, UK
Salvatore Caddemi, Italy
Piermarco Cannarsa, Italy
Jose E. Capilla, Spain
Carlo Cattani, Italy
Marcelo M. Cavalcanti, Brazil
Diego J. Celentano, Chile
Mohammed Chadli, France
Arindam Chakraborty, USA
Yong-Kui Chang, China
Michael J. Chappell, UK
Kui Fu Chen, China
Xinkai Chen, Japan
Kue-Hong Chen, Taiwan
Jyh-Horng Chou, Taiwan
Slim Choura, Tunisia
Cesar Cruz-Hernandez, Mexico
Swagatam Das, India
Filippo de Monte, Italy
Antonio Desimone, Italy
Yannis Dimakopoulos, Greece
Baocang Ding, China
Joao B. R. Do Val, Brazil
Daoyi Dong, Australia
B. Dubey, India
Horst Ecker, Austria
M. Onder Efe, Turkey
Elmetwally Elabbasy, Egypt
A. Elías-Zúñiga, Mexico
Anders Eriksson, Sweden
Vedat S. Erturk, Turkey
Moez Feki, Tunisia
Ricardo Femat, Mexico
Robertt A. Valente, Portugal
C. Fuerte-Esquivel, Mexico
Zoran Gajic, USA
Ugo Galvanetto, Italy
Furong Gao, Hong Kong
Xin-Lin Gao, USA
Behrouz Gatmiri, Iran
Oleg V. Gendelman, Israel
Didier Georges, France
Paulo B. Gonçalves, Brazil
Oded Gottlieb, Israel
Fabrizio Greco, Italy
Quang Phuc Ha, Australia
M. R. Hajj, USA
Tony S. W. Hann, Taiwan
Thomas Hanne, Switzerland
K. R. (Stevanovic) Hedrih, Serbia
M.I. Herreros, Spain
Wei-Chiang Hong, Taiwan
Jaromir Horacek, Czech Republic
Huabing Huang, China
Chuangxia Huang, China
Gordon Huang, Canada
Yi Feng Hung, Taiwan
Hai-Feng Huo, China
Asier Ibeas, Spain
Anuar Ishak, Malaysia
Reza Jazar, Australia
Zhijian Ji, China
Jun Jiang, China
J. J. Judice, Portugal
Tadeusz Kaczorek, Poland
Tamas Kalmar-Nagy, USA
Tomasz Kapitaniak, Poland
Hamid Reza Karimi, Norway
Metin O. Kaya, Turkey
Nikolaos Kazantzis, USA
Farzad Khani, Iran
K. Krabbenhoft, Australia
Ren-Jieh Kuo, Taiwan
Jurgen Kurths, Germany
Claude Lamarque, France
Usik Lee, Korea
Marek Lefik, Poland
Stefano Lenci, Italy
Roman Lewandowski, Poland
Shanling Li, Canada
Ming Li, China
Jian Li, China
Shihua Li, China
Teh-Lu Liao, Taiwan
Panos Liatsis, UK
Shueei M. Lin, Taiwan
Yi-Kuei Lin, Taiwan
Jui-Sheng Lin, Taiwan
Yuji Liu, China
Wanquan Liu, Australia
Bin Liu, Australia
Paolo Lonetti, Italy
V. C. Loukopoulos, Greece
Junguo Lu, China
Chien-Yu Lu, Taiwan
Alexei Mailybaev, Brazil
Manoranjan K. Maiti, India
O. D. Makinde, South Africa
R. Martinez-Guerra, Mexico
Driss Mehdi, France
Roderick Melnik, Canada
Xinzhu Meng, China
Yuri V. Mikhlin, Ukraine
G. Milovanovic, Serbia
Ebrahim Momoniat, South Africa
Trung Nguyen Thoi, Vietnam
Hung Nguyen-Xuan, Vietnam
Ben T. Nohara, Japan
Sotiris K. Ntouyas, Greece
Gerard Olivar, Colombia
Claudio Padra, Argentina
Bijaya Ketan Panigrahi, India
Francesco Pellicano, Italy
Matjaz Perc, Slovenia
Vu Ngoc Phat, Vietnam
M. do Rosário Pinho, Portugal
A. Pogromsky, The Netherlands

Seppo Pohjolainen, Finland
Stanislav Potapenko, Canada
Sergio Preidikman, USA
Carsten Proppe, Germany
Hector Puebla, Mexico
Justo Puerto, Spain
Dane Quinn, USA
K. R. Rajagopal, USA
Gianluca Ranzi, Australia
Sivaguru Ravindran, USA
G. Rega, Italy
Pedro Ribeiro, Portugal
J. Rodellar, Spain
R. Rodriguez-Lopez, Spain
A. J. Rodriguez-Luis, Spain
Ignacio Romero, Spain
Hamid Ronagh, Australia
Carla Roque, Portugal
Rubén R. García, Spain
Manouchehr Salehi, Iran
Miguel A. Sanjuán, Spain
Ilmar F. Santos, Denmark
Nickolas S. Sapidis, Greece
E. J. Sapountzakis, Greece
Bozidar Sarler, Slovenia
Andrey V. Savkin, Australia
Massimo Scalia, Italy
Mohamed A. Seddeek, Egypt
A. P. Seyranian, Russia
Leonid Shaikhet, Ukraine

Cheng Shao, China
Bo Shen, Germany
Jian-Jun Shu, Singapore
Zhan Shu, UK
Dan Simon, USA
Luciano Simoni, Italy
Grigori M. Sisoiev, UK
Christos H. Skiadas, Greece
Davide Spinello, Canada
Sri Sridharan, USA
Rolf Stenberg, Finland
Changyin Sun, China
Jitao Sun, China
Xi-Ming Sun, China
Andrzej Swierniak, Poland
Yang Tang, Germany
Allen Tannenbaum, USA
Cristian Toma, Romania
Irina N. Trendafilova, UK
Alberto Trevisani, Italy
Jung-Fa Tsai, Taiwan
K. Vajravelu, USA
Victoria Vampa, Argentina
Josep Vehi, Spain
Stefano Vidoli, Italy
Xiaojun Wang, China
Dan Wang, China
Youqing Wang, China
Yongqi Wang, Germany
Cheng C. Wang, Taiwan

Moran Wang, China
Yijing Wang, China
Gerhard-Wilhelm Weber, Turkey
J. A. S. Witteveen, The Netherlands
Kwok-Wo Wong, Hong Kong
Ligang Wu, China
Zhengguang Wu, China
Gongnan Xie, China
Wang Xing-yuan, China
Xi Frank Xu, USA
Xuping Xu, USA
Jun-Juh Yan, Taiwan
Xing-Gang Yan, UK
Suh-Yuh Yang, Taiwan
Mahmoud T. Yassen, Egypt
Mohammad I. Younis, USA
Bo Yu, China
Huang Yuan, Germany
S.P. Yung, Hong Kong
Ion Zaballa, Spain
Ashraf M. Zenkour, Saudi Arabia
Jianming Zhan, China
Xu Zhang, China
Yingwei Zhang, China
Lu Zhen, China
Liancun Zheng, China
Jian Guo Zhou, UK
Zexuan Zhu, China
Mustapha Zidi, France

Contents

Mathematical Methods Applied to Digital Image Processing, Yi-Hung Liu, Chung-Hao Chen, and Paul C.-P. Chao
Volume 2014, Article ID 480523, 4 pages

A New Dataset Size Reduction Approach for PCA-Based Classification in OCR Application, Mohammad Amin Shayegan and Saeed Aghabozorgi
Volume 2014, Article ID 537428, 14 pages

A Global Multilevel Thresholding Using Differential Evolution Approach, Kanjana Charansiriphaisan, Sirapat Chiewchanwattana, and Khamron Sunat
Volume 2014, Article ID 974024, 23 pages

Fuzzy-Based Segmentation for Variable Font-Sized Text Extraction from Images/Videos, Samabia Tehsin, Asif Masood, Sumaira Kausar, and Fahim Arif
Volume 2014, Article ID 389547, 10 pages

Hiding a Covert Digital Image by Assembling the RSA Encryption Method and the Binary Encoding Method, Kuang Tsan Lin and Sheng Lih Yeh
Volume 2014, Article ID 603659, 8 pages

The Nonlocal Sparse Reconstruction Algorithm by Similarity Measurement with Shearlet Feature Vector, Wu Qidi, Li Yibing, Lin Yun, and Yang Xiaodong
Volume 2014, Article ID 586014, 8 pages

Robust Homography Estimation Based on Nonlinear Least Squares Optimization, Wei Mou, Han Wang, and Gerald Seet
Volume 2014, Article ID 897050, 8 pages

Neighborhood Preserving Convex Nonnegative Matrix Factorization, Jiang Wei, Li Min, and Zhang Yongqing
Volume 2014, Article ID 154942, 8 pages

Moving Object Detection and Shadow Removing under Changing Illumination Condition, Jinhai Xiang, Heng Fan, Honghong Liao, Jun Xu, Weiping Sun, and Shengsheng Yu
Volume 2014, Article ID 827461, 10 pages

Low-Resolution Tactile Image Recognition for Automated Robotic Assembly Using Kernel PCA-Based Feature Fusion and Multiple Kernel Learning-Based Support Vector Machine, Yi-Hung Liu, Yu-Tsung Hsiao, Wei-Teng Cheng, Yan-Chen Liu, and Jui-Yiao Su
Volume 2014, Article ID 497275, 11 pages

An Improved Adaptive Deconvolution Algorithm for Single Image Deblurring, Hsin-Che Tsai and Jiunn-Lin Wu
Volume 2014, Article ID 658915, 11 pages

A Support Vector Data Description Committee for Face Detection, Yi-Hung Liu, Yung Ting, Shian-Shing Shyu, Chang-Kuo Chen, Chung-Lin Lee, and Mu-Der Jeng
Volume 2014, Article ID 478482, 9 pages

Optimal $O(1)$ Bilateral Filter with Arbitrary Spatial and Range Kernels Using Sparse Approximation, Shengdong Pan, Xiangjing An, and Hangen He
Volume 2014, Article ID 289517, 11 pages

Fast Total-Variation Image Deconvolution with Adaptive Parameter Estimation via Split Bregman Method, Chuan He, Changhua Hu, Wei Zhang, Biao Shi, and Xiaoxiang Hu
Volume 2014, Article ID 617026, 9 pages

A Fast Iterative Pursuit Algorithm in Robust Face Recognition Based on Sparse Representation, Zhao Jian, Huang Luxi, Jia Jian, and Xie Yu
Volume 2014, Article ID 683494, 11 pages

Robust Calibration of Cameras with Telephoto Lens Using Regularized Least Squares, Mingpei Liang, Xinyu Huang, Chung-Hao Chen, Gaolin Zheng, and Alade Tokuta
Volume 2014, Article ID 689429, 9 pages

A Block-Based Regularized Approach for Image Interpolation, Li Chen, Xiaotong Huang, and Jing Tian
Volume 2014, Article ID 953476, 8 pages

An Improved Metric Learning Approach for Degraded Face Recognition, Guofeng Zou, Yuanyuan Zhang, Kejun Wang, Shuming Jiang, Huisong Wan, and Guixia Fu
Volume 2014, Article ID 724978, 10 pages

Sensor Selection and Integration to Improve Video Segmentation in Complex Environments, Adam R. Reckley, Wei-Wen Hsu, Chung-Hao Chen, Gangfeng Ma, and E-Wen Huang
Volume 2014, Article ID 654790, 14 pages

Efficient Stereo Matching with Decoupled Dissimilarity Measure Using Successive Weighted Summation, Cheng-Tao Zhu, Yau-Zen Chang, Huai-Ming Wang, Kai He, Shih-Tseng Lee, and Chung-Fu Lee
Volume 2014, Article ID 127284, 9 pages

Poissonian Image Deconvolution via Sparse and Redundant Representations and Framelet Regularization, Yu Shi, Houzhang Fang, and Guoyou Wang
Volume 2014, Article ID 917040, 10 pages

Rendering Distortion Estimation Model for 3D High Efficiency Depth Coding, Qiuwen Zhang, Liang Tian, Lixun Huang, Xiaobing Wang, and Haodong Zhu
Volume 2014, Article ID 940737, 7 pages

Fast Threshold Selection Algorithm of Infrared Human Images Based on Two-Dimensional Fuzzy Tsallis Entropy, Dong-xue Xia, Chun-gui Li, and Shu-hong Yang
Volume 2014, Article ID 308164, 10 pages

Image Denoising Using Total Variation Model Guided by Steerable Filter, Wenxue Zhang, Yongzhen Cao, Rongxin Zhang, and Yuanquan Wang
Volume 2014, Article ID 423761, 11 pages

2D Barcode Image Decoding, Jeng-An Lin and Chiou-Shann Fuh
Volume 2013, Article ID 848276, 10 pages

Editorial

Mathematical Methods Applied to Digital Image Processing

Yi-Hung Liu,¹ Chung-Hao Chen,² and Paul C.-P. Chao³

¹ Department of Mechanical Engineering, Chung Yuan Christian University, Chungli 32023, Taiwan

² Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529, USA

³ Department of Electrical Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan

Correspondence should be addressed to Yi-Hung Liu; lyh@cycu.edu.tw

Received 5 March 2014; Accepted 5 March 2014; Published 22 April 2014

Copyright © 2014 Yi-Hung Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Digital image processing (DIP) is an important research area since it spans a variety of applications. Although over the past few decades there has been a rapid rise in this field, there still remain issues to address. Examples include image coding, image restoration, 3D image processing, feature extraction and analysis, moving object detection, and face recognition. To deal with these issues, the use of sophisticated and robust mathematical algorithms plays a crucial role. The aim of this special issue is to provide an opportunity for researchers to publish their latest theoretical and technological achievements in mathematical methods and their various applications related to DIP. This special issue covers topics related to the development of mathematical methods and their applications. It has a total of twenty-four high-quality papers covering various important topics in DIP, including image preprocessing, image encoding/decoding, stereo image reconstruction, dimensionality and data size reduction, and applications.

2. Image Preprocessing

One of the topics covered in this special issue is related to the image preprocessing methods, including segmentation, thresholding, denoising, image interpolation, calibration of cameras, and image restoration. There are twelve papers related to this topic, occupying half of the papers included in this special issue. The paper entitled “An improved adaptive deconvolution algorithm for single image deblurring” by H.-C. Tsai and J.-L. Wu proposes an adaptive regularization

method, which can preserve image details and suppress ringing artifacts simultaneously. Their results carried out on synthesized and real images show that their method can restore latent image with much fewer ringing and favors the sharp edges. S. Pan et al. propose the paper entitled “Optimal $O(1)$ bilateral filter with arbitrary spatial and range kernels using sparse approximation.” In this paper, an approximating arbitrary spatial kernel using a fixed number of boxes is presented. The multiple-box spatial kernel can be applied in many $O(1)$ acceleration schemes in addition to the histogram-based one. Their results show that the proposed method has better accuracy in approximating the bilateral filter with Gaussian spatial kernel, compared with previous histogram-based methods.

Then, the paper titled “Fast total-variation image deconvolution with adaptive parameter estimation via split Bregman method” by C. He et al. studies how to balance the total-variation (TV) regularization and data fidelity, which has considered a challenging issue in image restoration. Their proposed method extends the classical split Bregman method to a new fast algorithm, which simultaneously estimates the regularization parameter and restores the blurred image. The regularization parameter is refreshed conveniently in a closed form according to Morozov’s discrepancy principle. Another paper of this topic is “Robust calibration of cameras with telephoto lens using regularized least squares” by M. Liang et al. This paper has two main contributions. First, they present a first-order error analysis that shows the relation between focal length and estimation uncertainties of camera parameters. To our knowledge, this error analysis with respect to focal length has not been studied in the area of camera calibration.

Second, they propose a robust algorithm to calibrate the camera with a long focal length without using additional devices: the covariance of camera parameters can be reduced greatly.

The next paper of this topic is “*A block-based regularized approach for image interpolation*” by L. Chen et al. This paper studies how to render a high-resolution image from a low-resolution image. Classical interpolation techniques estimate the missing pixels from the surrounding pixels based on a pixel-by-pixel basis. In contrast, this paper presents a new efficient algorithm for image interpolation based on regularization theory, which formulates the interpolation problem as an optimization problem in which the cost function consists of a data fidelity term and regularization functional. The computational cost of this algorithm is further reduced by incorporating Kronecker product and singular value decomposition. The effect of regularization on the interpolation results is analyzed in this paper. Next, the paper entitled “*Sensor selection and integration to improve video segmentation in complex environments*” coauthored by A. R. Reckley et al. proposes a technique to integrate spatiotemporal signatures of an object of interest from different sensing modalities into a video segmentation method in order to improve object detection and tracking in dynamic, complex scenes. Their proposed algorithm utilizes the dynamic interaction information between the object of interest and background to differentiate between mistakenly segmented components and the desired component. S. Tehsin et al. proposed the paper entitled “*Fuzzy-based segmentation for variable font-sized text extraction from images/videos*,” which addresses the issue of text localization and detection by using a novel fuzzy-based segmentation method.

Moreover, Poissonian image deconvolution is a key issue in various applications, such as astronomical imaging, medical imaging, and electronic microscope imaging. To address this issue, Y. Shi et al. propose the paper “*Poissonian image deconvolution via sparse and redundant representations and framelet regularization*”. In this paper, an approach that combines analysis with synthesis methods is proposed. The aim is to address the Poissonian image deconvolution problem by minimizing an energy functional. The minimization problem can be efficiently solved by the split Bregman technique. D.-X. Xia et al. propose the paper entitled “*Fast threshold selection algorithm of infrared human images based on two-dimensional fuzzy Tsallis entropy*.” In this paper, a fast thresholding method of infrared human images based on two-dimensional fuzzy Tsallis entropy is introduced. First, to address the fuzziness of infrared image, the fuzzy Tsallis entropy of objects and that of background are defined, respectively, according to probability partition principle. Next, this newly defined entropy is extended to two dimensions to make good use of spatial information to deal with the noise in infrared images, and correspondingly a fast computation method of two-dimensional fuzzy Tsallis entropy is put forward to reduce its computation complexity.

Another paper to this set is “*Image denoising using total variation model guided by steerable filter*” by W. Zhang et al. This paper proposes an adaptive total variation (TV) model by introducing the steerable filter into the TV-based diffusion

process for image filtering. The proposed model can provide a better image processing tool which enables noise removal, edge-preserving, and staircase suppression. The paper also that belongs to this topic is the one entitled “*A global multilevel thresholding using differential evolution approach*” by K. Charansiriphaisan et al. The optimal Otsu threshold values of multilevel image thresholding are necessary for some applications and a global search algorithm is required. Previously, differential evolution (DE) algorithm has been used successfully for solving this problem. However, the difficulty of problem grows exponentially when the number of thresholds increases. To overcome this difficulty, an improved DE using new mutation strategy is proposed in this paper. The last paper of this topic is related to image restoration. W. Qidi, et al. propose the paper entitled “*The nonlocal sparse reconstruction algorithm by similarity measurement with Shearlet feature vector*,” which supplies a nonlocal sparsity reconstruction algorithm with similarity measurement to improve the accuracy of conventional methods of image restoration. To improve the performance of restoration results, this paper proposes two schemes to dictionary learning and sparse coding, respectively. In the part of the dictionary learning, the similarity between patches from degraded image by constructing the Shearlet feature vector is measured. In the part of sparse coding, an optimal objective function with the coding residual item is performed, which can suppress the residual between the estimate coding and true sparse coding. The derivation of self-adaptive regularization parameter in optimization under the Bayesian framework is also provided, which can make the performance better.

3. Stereo Image Reconstruction

In this special issue, there are three papers related to the topic of stereo image reconstruction. The homography between image pairs is normally estimated by minimizing a suitable cost function, given a set of 2D keypoint correspondences. The correspondences are typically established using descriptor distance of keypoints. However, the correspondences are often incorrect due to ambiguous descriptors which can introduce errors into following homography computing step. There have been numerous attempts to filter out these erroneous correspondences, but it is unlikely to always achieve perfect matching. To deal with this problem, W. Mou et al. propose the paper entitled “*Robust homography estimation based on nonlinear least squares optimization*,” in which a nonlinear least squares optimization approach is presented to compute homography such that false matches have no or little effect on computed homography.

Moreover, developing matching algorithms from stereo image pairs to obtain correct disparity maps for 3D reconstruction has also been the focus of intensive research. A constant computational complexity algorithm to calculate dissimilarity aggregation in assessing disparity based on separable successive weighted summation (SWS) among horizontal and vertical directions was proposed but still not satisfactory. To address this issue, C.-T. Zhu et al. propose the paper entitled “*Efficient stereo matching with decoupled*

dissimilarity measure using successive weighted summation,” which presents a novel method enabling decoupled dissimilarity measure in the aggregation, further improving the accuracy and robustness of stereo correspondence. The aggregated cost is also used to refine disparities based on a local curve-fitting procedure.

Then, the paper entitled “*Rendering distortion estimation model for 3D high efficiency depth coding*” by Q. Zhang et al. proposes a modified distortion estimation model based on view rendering distortion. This model can be applied to the high efficiency video coding (HEVC) rate distortion cost function process for rendering view quality optimization. Their experimental results on various 3D video sequences show that this model can provide about 31% BD-rate savings in comparison with HEVC simulcast and 1.3 dB BD-PSNR coding gain for the rendered view.

4. Image Encoding and Decoding

Two papers published in this special issue are focused on image encoding and decoding. The paper entitled “*2D barcode image decoding*” coauthored by J.-A. Lin and C.-S. Fuh studies how to decode various quick response (QR) code images efficiently and accurately. This paper revises the traditional decoding procedure by proposing a serial of carefully designed preprocessing methods. The decoding procedure consists of image binarization, QR code extraction, perspective transformation and resampling, and error correction. By these steps, different types of QR code images can be recognized accurately. K. T. Lin and S. L. Yeh propose the paper entitled “*Hiding a covert digital image by assembling the RSA encryption method and the binary encoding method,*” which integrates the Rivest-Shamir-Adleman encryption method and the binary encoding method to form a hybrid hiding method that can hide a covert digital image into a dot-matrix holographic image. The pixels of the dot-matrix holographic image contain seven groups of codes used for reconstructing the covert image, and the reconstructed covert image derived from the dot-matrix holographic image and the original covert image are exactly the same.

5. Dimensionality and Dataset Size Reduction

There are two papers in this topic. In image recognition, feature extraction plays a critical role. Feature extraction typically suffers from the problem of high dimensionality. Therefore, the issue of dimension reduction is of primary importance. The convex nonnegative matrix factorization (CNMF) is a variation of nonnegative matrix factorization (NMF) in which each cluster is expressed by a linear combination of the data points and each data point is represented by a linear combination of the cluster centers. When there exists nonlinearity in the manifold structure, both NMF and CNMF are incapable of characterizing the geometric structure of the data. The paper “*Neighborhood preserving convex nonnegative matrix factorization*” by J. Wei et al. introduces a neighborhood preserving convex nonnegative

matrix factorization (NPCNMF), which imposes an additional constraint on CNMF that each data point can be represented as a linear combination of its neighbors. This method is able to reap the benefits of both nonnegative data factorization and the purpose of manifold structure. The feasibility and effectiveness of NPCNMF are verified on several standard data sets with promising results.

Another major problem of pattern recognition systems is due to the large volume of training datasets including duplicate and similar training samples. As a result, a large memory is required for saving data, as well as the high time complexity for training the algorithms. The paper entitled “*A new dataset size reduction approach for PCA-based classification in OCR application*” by M. A. Shayegan and S. Aghabozorgi reports on the use of a version of the Modified Frequency Diagram technique for dataset size reduction. In this new proposed technique, a big training dataset is rearranged and then sieved. The sieved training dataset along with automatic feature extraction/selection operation using Principal Component Analysis (PCA) is used in an optical character recognition (OCR) application. The experimental results obtained when using the proposed system on one of the biggest handwritten Arabic/Farsi numeral standard OCR datasets, Hoda, show about 97% accuracy in the recognition rate. The recognition speed increased by 2.28 times, while the accuracy decreased only by 0.7%.

6. Applications

In this special issue, five papers are related to DIP applications. Among the five papers, three are directly related to face detection and recognition. Y.-H. Liu et al. propose the paper entitled “*A support vector data description committee for face detection,*” which develops a one-class learning-based face detector called support vector data description (SVDD) committee, which consists of several SVDD members, each of which is trained on a subset of face patterns. Since nonfaces are not required in the training of the SVDD committee, face detection accuracy of SVDD committee is independent of the nonface training patterns. Experiments carried out on the extended MIT face dataset show that the proposed SVDD committee can achieve better face detection accuracy than the widely used SVM face detector. Z. Jian et al. propose the paper entitled “*A fast iterative pursuit algorithm in robust face recognition based on sparse representation.*” In this paper, a relatively fast pursuit algorithm for face recognition is proposed. The recognition rates on ORL database, Yale database, and FERET database are 95.5%, 93.87%, and 92.29%, respectively. The recognition performance under various levels of occlusion and corruption is also experimentally proved to be significantly enhanced. The paper “*An improved metric learning approach for degraded face recognition*” by G. Zou et al. also addresses the face recognition problem. To solve the matching problem of the elements in different data collections, an improved coupled metric learning approach is proposed in this paper. Experiments based on Yale and CAS-PEAL-RI face databases demonstrate that the proposed kernel coupled approach performs better in low-resolution

and fuzzy face recognition and can reduce the computing time.

In addition to face recognition, other problems are also challenging in real-world applications. J. Xiang et al. propose the paper entitled “*Moving object detection and shadow removing under changing illumination condition*” that addresses the problem of moving object segmentation by proposing a local intensity ratio model that is robust to illumination change. Y. H. Liu et al. propose the paper entitled “*Low-resolution tactile image recognition for automated robotic assembly using kernel PCA-based feature fusion and multiple kernel learning-based support vector machine.*” In this paper, a robust tactile sensing image recognition scheme is proposed, a kernel PCA feature fusion layer is performed to extract distinguishing features from a set of candidate features including Fourier descriptors and geometric features, and a multiple kernel learning support vector machine is used as the classifier. The proposed recognition scheme is able to achieve a high recognition rate of over 85% for the classification of 12 commonly used metal parts in industrial applications.

Acknowledgments

The guest editors would like to thank the authors and reviewers of the journal who worked on the publication of this special issue.

*Yi-Hung Liu
Chung-Hao Chen
Paul C.-P. Chao*

Research Article

A New Dataset Size Reduction Approach for PCA-Based Classification in OCR Application

Mohammad Amin Shayegan¹ and Saeed Aghabozorgi²

¹ *Image Processing and Pattern Recognition Research Lab, R&D Center, Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia*

² *Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia*

Correspondence should be addressed to Mohammad Amin Shayegan; mashaygan@siswa.um.edu.my

Received 25 August 2013; Revised 14 January 2014; Accepted 19 January 2014; Published 17 April 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 M. A. Shayegan and S. Aghabozorgi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A major problem of pattern recognition systems is due to the large volume of training datasets including duplicate and similar training samples. In order to overcome this problem, some dataset size reduction and also dimensionality reduction techniques have been introduced. The algorithms presently used for dataset size reduction usually remove samples near to the centers of classes or support vector samples between different classes. However, the samples near to a class center include valuable information about the class characteristics and the support vector is important for evaluating system efficiency. This paper reports on the use of Modified Frequency Diagram technique for dataset size reduction. In this new proposed technique, a training dataset is rearranged and then sieved. The sieved training dataset along with automatic feature extraction/selection operation using Principal Component Analysis is used in an OCR application. The experimental results obtained when using the proposed system on one of the biggest handwritten Farsi/Arabic numeral standard OCR datasets, Hoda, show about 97% accuracy in the recognition rate. The recognition speed increased by 2.28 times, while the accuracy decreased only by 0.7%, when a sieved version of the dataset, which is only as half as the size of the initial training dataset, was used.

1. Introduction

The emergence of the Big-Data issue has caused researchers to focus their attention on the size reduction and also dimensionality reduction of the data to save time and memory usage. Also, there is an increasing demand for employing various applications on limited-speed and limited-memory devices such as mobile phones and mobile scanners [1]. In this context, there is a pressing need to find efficient techniques for reducing the volume of data in order to decrease overall processing time and memory requirements.

A survey of the literature on large dataset issue reveals that two general approaches are used for dataset volume reduction—dimensionality reduction and size deduction. In the dimensionality reduction technique, the system will try to find and remove less important extracted features from the dataset samples. These techniques are widely employed

in different areas such as EMG signal feature reduction [2] and gene expression dataset reduction [3]. Specific examples of these techniques include Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Random Projection (RP), and Mean-Variance Approach [4, 5]. However, finding an optimal, effective, and robust feature set from a big initial extracted feature is usually a heuristic and inevitably a difficult task [6].

In the size reduction techniques, the system will try to reduce the number of objects or observations in a dataset. Such techniques find and remove two groups of samples from a dataset—samples far from a class centroid (outlier samples or support vector samples) [7–9] and samples near to each class centroid (e.g., using K-means clustering technique) [8, 10]. However, the samples near to a class centroid include important information about various characteristics of a class, and they are necessary to make the system model.

Also, the outlier and support vector samples are necessary to evaluate the system efficiency and functionality.

Currently, large amounts of indispensable data are available on paper. Converting the graphical-text data images into editable text documents (Optical Character Recognition (OCR)) is one of the most attractive branches in the pattern recognition (PR) domain that scientists have been faced with in recent years. Meanwhile, about more than one billion people worldwide use the Arabic, Farsi, and other similar alphabets as their native language [11]. About 30% of the world population and around 30 world languages use this group of alphabet sets as a base script for writing, and this makes it the second most-used alphabet set used in writing worldwide [12]. Hence, it is not surprising that these alphabets have received a lot of attention in the recent years. Concerning dataset size reduction in OCR applications, the PCA technique has been used to compress the features space in the numeral part of the CEDAR database [13], the MNIST database [14], and also the handwritten Tamil characters classes [15]. However, to the best of our knowledge, no research efforts have been undertaken regarding dataset size reduction for the Arabic/Farsi language.

This paper presents a new method to achieve the above purpose by using a Modified Frequency Diagram (MFD) matching technique and a new similarity measurement function, in particular, to reduce the number of samples in the training section of an Arabic/Farsi dataset. Using the MFD technique, a template is first generated for each class in the pattern space. A new similarity measurement function is then used to compute a similarity value for each training sample of a dataset corresponding to that class template. Thereafter, the samples in a specific class are sorted in descending order based on the calculated similarity values. Finally, the sorted training dataset is sieved at sampling rates of 1/2, 1/3, and 1/4, and the dataset size is reduced to 1/2, 1/3, or 1/4 of its original size. To investigate the efficiency of the proposed dataset reduction technique, the sieved training dataset along with PCA is used in the recognition stage of an OCR application to recognize handwritten Arabic/Farsi numerals. The results from a successful application on an OCR database have been reported, but what is more important to note is that this method, with some adjustment, can be used for other pictorial datasets.

The main difference between our proposed method and other available methods for dataset size reduction is that our model keeps some samples which are close to a class template, as well as those which are far from a class template, in the final reduced dataset version. Hence, it is possible to have appropriate samples for making a system model and also for assessing the system efficiency with a high degree of accuracy. Moreover, in this research, PCA is employed both for automatic feature extraction and feature reduction (selection) in OCR applications.

This paper contributes to the corpus of knowledge in dataset size reduction as follows: (1) it proposes a new method for dataset size reduction to speed up system training and testing and (2) it is the first successful effort to be reported on the use of PCA for automating feature extraction, in

addition to reducing dimensionality (feature reduction) in OCR applications.

This paper is organized as follows. Section 2 discusses the background of the research topic and introduces works related to the topic. Section 3 presents the proposed size reduction technique, the research method, and the experimental procedures. Section 4 presents the results of the experiments and the analysis, and finally, Section 5 concludes the paper.

2. Background and Related Works

2.1. Principal Component Analysis (PCA). The Principal Component Analysis (PCA) technique is a classical statistical linear transform which has been widely used for different PR applications such as data compression, face recognition, and character recognition [6]. It has been applied to find important patterns in high-dimensional input data. It converts a correlated feature space into a noncorrelated feature space. In the new space, features are reordered in decreasing variance values—based on the generated eigenvectors from the training data—such that the first transformed feature accounts for the most variability in the data.

PCA can briefly be described as follows. Let B be a $N \times N$ pixels binary image as a random vector population. A 2D image B is converted to a 1D vector X by concatenating all rows of the image in order as follows:

$$X = (x_1, \dots, x_n)^T, \quad (1)$$

where X_i is the pixel values of row i . If μ_x is the mean of X , then the covariance matrix of that population is C_x :

$$C_x = E \{ (x - \mu_x)(x - \mu_x)^T \}. \quad (2)$$

In order to normalize the data values in each dimension, data is subtracted from the corresponding mean. This changes the mean of each dimension to zero. The components of C_x represent the covariance between the random variable components x_i and x_j . The covariance matrix C_x is a square matrix, and therefore its eigenvalues and eigenvectors can be calculated. The eigenvectors are generally perpendicular to each other. The eigenvector with the biggest corresponding eigenvalue is the most significant representative data and is considered to be the first most significant principal component. The eigenvector with the second biggest corresponding eigenvalue is considered to be the second most significant principal component, and so on. Therefore, by sorting the eigenvalues in descending order, the most important eigenvectors as the most significant representation data are found. In this way, the pattern space dimension can be reduced.

However, the computation of PCA requires eigenvalue decomposition of the covariance matrix of the feature vectors with around $O(d^3 + d^2n)$ computations, where n is the number of samples and d is the dimensionality of the feature space [16]. This powerful technique, however, is time consuming and is usually employed for feature extraction/selection operations on small scale datasets.

2.2. Feature Extraction and Feature Selection. Feature extraction (FE) is a task to detect and extract the maximum amount of the desired attributes from the input data. Features are the information that is fed to the recognizer to build a system model [17]. They should be insensitive to irrelevant variability in the input as much as possible, should be limited in number to permit effective computation of discriminant functions, and should not be similar, redundant, or repetitive. Usually, extracting appropriate and robust features is a difficult task in an OCR system, like in other PR applications.

Various features are computed in the feature extraction module in an OCR system. The features are categorized into global transformations such as Fourier, structural features such as ascenders and descenders, statistical features such as moments, and template matching and correlation.

Many different features can be found or calculated for each pattern in a PR system. Some of the features, however, might correspond to very small details of the pattern, or some of them might be a combination of other features (nonorthogonal features), while others might not play any effective role in the recognition stage. Irrelevant or redundant features may degrade the recognition results and significantly reduce the speed of the learning algorithms. Hence, using all extracted features does not always produce the desired results and could also increase the time complexity of the recognition process [18]. Therefore, following the feature extraction process, another important process—feature selection (FS)—is involved and this process can reduce the problem dimensionality. FS is typically a searching technique for finding an optimal subset with m features out of the original M features.

The first category of feature selection methods is Sequential Backward Selection (SBS). In this approach, features are deleted one by one and system performance is measured to determine the feature performance. However, it is very important to find the correct order of deleting the features one by one. This is because the derived efficiencies of a system after deleting features A , B , and C are not the same as its derived efficiencies after deleting the features in the order A , C , and B , or B , C , and A , and so on [19].

The second category of feature selection methods comprises the random search methods such as Genetic Algorithms (GA). The GA methods select chromosomes (features) with the best recognition percentage, one by one, and move this chromosome to the next stage. However, it is possible that when a good characteristic feature is combined with another feature, the overall performance will not be as good as the performance of each of the features, separately. Azmi et al. [18] used GA in an OCR system for recognizing handwritten texts. Initially, there were 81 features and the recognition rate was 77%. After applying GA, the number of features was reduced to 55 and the recognition rate improved from 77% to 80%. Kheyrikhah and Rahmanian [20] employed GA to optimize the number of initial extracted features in a recognition system for handwritten digits. They found that all the extracted features are not useful in classification, and they also reduce recognition accuracy and increase the system's learning time. Their system was able to reduce the number of features from 48 to 30 and increase the recognition rate from

75% to 94%, but the elapsed time for these improvements was significant.

The third category of feature selection methods is represented by the Principal Component Analysis (PCA). PCA transforms data into a new space where the features are uncorrelated. In the new space, features are reordered in decreasing variance value such that the first transformed feature accounts for the most variability in data. Hence, PCA is able to overcome the problem of high dimensionality and colinearity [6]. It is obvious that handwritten digits and characters have a wide variety of writing styles; hence, handwritten texts are placed in the high-dimension input space category. It means that PCA can be an effective tool for attribute reduction in OCR applications.

2.3. Dataset Size Reduction. There have been researches on finding methods to reduce dataset size in order to decrease the overall processing time in PR systems. Urmanov et al. [21] first calculated an original decision boundary equation D for each class of patterns. They then calculated the distance between each sample x in each class C and the original decision boundary equation D . Next, without using sample x in the same class C , they calculated a new decision boundary and the new distance with respect to this new decision boundary. If the pairs of old and new decision boundaries are very similar, sample x is considered worthless and is removed from the dataset.

Zhongdong et al. [7] attempted to reduce dataset volume by finding support vectors. They found the samples of each class near the boundary spaces and then calculated the distance of the found samples of each class from other classes. Finally, they considered the nearest couple of samples from any two classes as the most important, while the other samples of each class were removed from the classes and dataset. This approach was also used by Hara and Nakayama [9].

Vishwanathan and Murty [8] first used multicategory proximal SVMs to categorize training system prototypes and then found different boundaries for separating different classes of clusters. They removed not only samples which are generally close to the class boundaries but also the typical patterns that are far from the class boundaries.

For each pair of samples from two different classes, Javed et al. [22] plotted a sphere for each pair of samples, such that those two samples are put on two sides of the sphere diameter. If none of the other samples inside these two classes are within the sphere volume, these samples are nearest to each other from these two classes. Therefore, these samples are support vectors and will be inserted into the final dataset.

Boucheham [23] introduced the method of recursive Piecewise Linear Approximation (PLA) and sequential PLA for data reduction to speed up time series comparison. It is usually possible for printed characters to be considered as a time series, but the wide variation of styles of handwritten characters rules out any possibility of considering handwritten template as time series, and therefore the method mentioned is not applicable for handwritten documents.

Cervantes et al. [24] first obtained a sketch from the distribution of available classes with a small number of training samples, and they then identified existing support vectors in this limited dataset. Their proposed system is trained to find samples near the boundary between classes, and then other important samples are found and added to the final dataset.

In summary, it is found that the majority of the algorithms mentioned can be divided into two general groups as follows.

- (a) The first group of algorithms tries to find and delete support vector samples (SVs) in all classes [8]. These samples are usually the samples which are far from the class center and near to a class boundary. It is usual for a recognition system to justify or classify such samples wrongly. However, one of the main criteria for evaluating system efficiency and measuring the power of a PR system is the correct recognition of these SVs and outlier samples. Hence, it is not a good strategy to delete these patterns from the initial dataset in order to achieve dataset size reduction.
- (b) The second group of algorithms removes the samples near the centers of the classes from the initial training dataset to create a short final dataset version [7–9, 22, 24]. However, these samples include highly valuable information about a specific class that is needed for making a system model in the training phase of a PR system.

2.4. PCA-Based Classification. The Principal Component Analysis (PCA) technique is usually used for feature reduction, but sometimes it is also utilized for automatic feature extraction operation in various PR systems.

Zhang et al. [14] introduced a multimodal approach for reducing the features' dimensions in an OCR system and tested their approach on the numeral part of the MNIST dataset. They employed PCA for feature compression and succeeded in reducing the CPU time for classification. Kim et al. [13] modeled each digit class of the UCI dataset to several components and used a mixture model of PCA techniques to move extracted components into a decorrelated feature space. Based on the results obtained, the PCA mixture model outperforms other methods such as k -NN in terms of accuracy. In 2004, Deepu et al. [15] employed PCA for dimensionality reduction in an online OCR application for Tamil handwritten characters. They found that the modified version of the orthogonal distance classifier performs better when compared to the k -NN classifier. The novelty of this work is that it is language independent, and the proposed method can also be used for other language scripts. In 2005, Mozaffari et al. [25] used PCA technique to reduce the number of fractal code features to 240 in a handwritten zip code recognition system. They succeeded in improving the recognition rate from 86.3% to 90.6%. Ziaratban et al. [26] extracted a set of feature points—including terminal, two-way, and three-way branch points—from the skeletons of characters. Finally, each skeleton was decomposed into some primitives, which are curved lines between any two successive feature points. Since the number of primitives

varies from character to character, they used a PCA algorithm to reduce and equalize the length of the feature vectors. They achieved 93.15% recognition rate using a dataset with 7,647 test samples. To recognize handwritten Arabic isolated letters, Abandah et al. [6] extracted 95 features from input images. The PCA technique was then used for feature reduction, and only the first 40 features were selected. They achieved an average accuracy of 87%, in the best case. El-Glaly and Quek [19] extracted four sets of features, S1, S2, S3, and S4, from input data, for the use in an Arabic OCR system. They trained the system with the four feature sets, separately. These sets were then processed using the PCA algorithm, and PCA rearranged the features based on their importance which was identified by the recognition system. The results show that feature X in rank 23 in set S3 took rank 7 in set S1, and so on. The results of experiment indicate that if feature X is deleted merely to achieve feature reduction, it may give rise to serious errors in the final results.

2.5. Farsi OCR. This research was conducted specifically on the use of OCR technique for recognition of handwritten Arabic/Farsi digits. From a review of the relevant literature, some of the methods for recognizing handwritten Arabic/Farsi numerals are described briefly in the following sections.

Mowlaei and Faez [27] extracted a 64-dimensional feature vector of Harr wavelet coefficients for recognizing handwritten Farsi digits. They achieved 92.44% recognition rate of 3,840 digits by using an SVM classifier with an exponential RBF kernel in one-to-other mode.

In 2003, Sadri et al. [28] used SVM classifiers for recognizing the digits part of the Farsi-CENPARMI database. They obtained four different views for any image from four main directions by counting the number of background pixels between the border and outer boundary of any image, and finally they created a 64-dimensional feature vector for each image. Using SVMs with RBF kernel, a 94.14% recognition rate was achieved. To compare the classifiers, they employed an MLP-NN classifier with two hidden layers. This classifier achieved an accuracy rate of 91.25%, and this is lower than that obtained using the SVM classifier.

In 2004, Mozaffari et al. [29] proposed a new method for recognizing handwritten isolated Farsi characters and numerals for handling mail codes and city names in the post ministry of Iran. They extracted a 64-dimensional set of fractal codes, as well as wavelet transform coefficients. They employed an SVM with an exponential RBF kernel as a classifier. The final results for both feature sets were almost similar and a recognition rate of 91.3% was achieved. However, the system recognition speed, when it uses wavelet coefficients as features, is 25 times higher than the system which uses fractal codes as features.

Soltanzadeh and Rahmati [30] used the outer profile of digit images at multiple orientations—top, down, left, right, diagonal, off-diagonal, and so on—as the main features. They also used the normalizing crossing counts and projection histograms of any image as complementary features. The total number of features was $32 * n + 1$, where n is the number of orientations for calculating the outer profiles. They

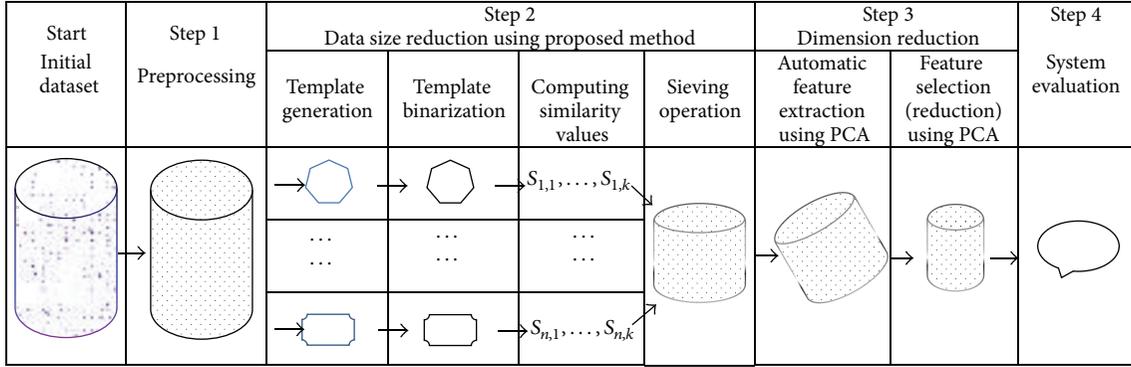


FIGURE 1: The proposed model for dataset (size and dimension) reduction.

employed an SVM classifier with a polynomial kernel in one experiment, and an RBF kernel in another experiment in the one-rest method. The best result they achieved was 99.57% recognition rate using eight orientation profiles and the RBF kernel.

Ziaratban et al. [31] extracted some language-based features including the position and the amount of the best-occurred matching in the horizontal and vertical coordinates for handwritten Farsi digit recognition. They tested the system on a database which contains 6,000 samples for training and 4,000 samples for testing, using an NN-MLP as a classifier, and successfully achieved 97.65% recognition rate.

Enayatifar and Alirezanejad [32] categorized images of digits into two groups: the first group includes digits 1, 2, 3, 4, 6, and 9 and the second group includes digits 0, 5, 7, and 8, based on the similarity of their skeletons. In the feature extraction stage, they divided the image of a digit into 24 frames and for each frame they calculated pixel accumulation and direction as features. Hence, a 48-dimensional feature vector was created for each digit. For the recognition operation, an MLP-NN classifier with 50 neurons in the hidden layer was employed. The best recognition rate achieved was 94.30%.

3. The Proposed Method

The existing methods already proposed for dataset size reduction generally try to find the boundary points (support vectors) between different classes in a pattern space. However, our proposed method adopts a completely different approach. Figure 1 depicts the general structure of the proposed model. The operations mentioned are described in the following sections.

3.1. Preprocessing. The performance of an OCR system depends very much upon the quality of the original data. In this context, we took into consideration that the proposed algorithm should be nonsensitive with respect to the scaling, rotation, and transformation of patterns. Hence, some important preprocessing operations, such as noise removal, dimension normalization, and slant correction using common powerful techniques, are first performed on the samples.

We applied a median filter with a 3×3 window and also morphological opening and closing operators using dilation and erosion techniques for noise removal. The image size was normalized without making any changes to the image aspect ratio, and as a result, the width or height (or both) was changed to 50 pixels and the image was located in the center of a 50×50 pixels bounding box.

The body in every Arabic/Farsi digit is constructed using only one component. Thus, after the preprocessing operations, if it is found that there is still more than one group of connected pixels in the image of the digit, the extra blocks are considered as noise or separate components of the initial image. To find and remove the rest of the noise, the pen width is estimated using three different methods, and then the average of those values is considered to be the final pen width. To achieve this, we compute the following:

- (a) the mode of the image vertical projection,
- (b) $(\text{the value of image density}) / (\text{the number of image skeleton pixels})$,
- (c) $\{(\text{the value of image density}) / (\text{the number of image outer profile pixels})\} * 2$.

The results from the experiments show that the average of three values is a more accurate estimate of the pen width than each of the values alone. After finding the pen width, all small components with a pixel density that is less than two times of the pen width can be considered to be noise and can be deleted from the input image. The threshold 2 was obtained experimentally. The rest of the connected components are considered as broken parts of the digit image.

In order to connect the broken image segments together, we used a new approach. By using connected component analysis, we named the biggest available part as the main part **M** of the image. The outer contour of the main part **M** was then extracted and the coordinates of its pixels were saved in array **MAIN**. Thereafter, for all of the rest secondary components **S_i** (which are smaller than the main part **M**), we found the outer contour and saved the pixels coordinate of those outer contours in another array **SEC**. Then, we computed the Euclidean distance between all elements of array **MAIN** with all elements of array **SEC**. The smallest value of the computed distance indicates the shortest path

```

while (there is another secondary component in input image) do
{
  find outer contour of the main part M;
  save the pixels coordinate of M in array MAIN;
  repeat
  {
    find outer contour of an image secondary part S;
    save the pixels coordinate of S in array SEC;
  } until (there is not another secondary parts in image);
  for (each pixel A in array MAIN)
  {
    for (each pixel B in array SEC)
    {
      compute the distance d between pixels A and B;
      save (d, coordinate of pixel A, coordinate of pixel B) in array D;
    }
  }
  d_min = smallest value d in array D;
  A_min = coordinate of pixel A, corresponding to d_min;
  B_min = coordinate of pixel B, corresponding to d_min;
  draw (a straight line with pen_width thickness from A_min to B_min);
}

```

ALGORITHM 1: The proposed procedure to connect the broken image segments.

between contour **M** and one of the secondary contours S_k . Finally, we drew a line with thickness equal to estimated pen width along the shortest path between **M** and S_k . As a result, the main part **M** is connected to a secondary part S_k . This process was repeated until there is not another secondary component. A new version of main part **M** is used in each iteration of the algorithm, because one secondary part is connected to the old version of main part **M** to make the new version of main part. Algorithm 1 demonstrates the pseudocode for this process.

We applied the proposed method on the digits part of the Hoda dataset [33] to connect the broken parts of the images of the digits. The results were encouraging as we were able to achieve 97.16% successful connections. Figure 2 shows an example of the above-mentioned preprocessing operations on two sets of digits from our training dataset.

The method proposed by Hanmandlu et al. [34] was used to correct the slant angle of each image. First, an image is divided into upper and lower halves. The centers of mass points for these two parts are then calculated. The slope of a line which connects these two mass point centers is considered to be the slant angle and the image is rotated in the reverse direction of this value. Figure 3 shows an example of slant correction.

3.2. Data Size Reduction

3.2.1. Template Generation for Each Class. By using all the preprocessed samples of each class, the Frequency Diagram (FD) (density grid) was first computed by calculating the number of occurrences of pixel "1" in the coordinates (x, y) for all available samples in a special class [35]. The modified version of FD (MFD) was defined by Khosravi and Kabir [33].

They increased the pixel density variable D_i by 1 unit, if the pixel in coordinate (x, y) of an input sample is "1." In order to have a more accurate description of pixel density in each class, they decreased the pixel density variable D_i by 1 unit, if the pixel in coordinate (x, y) of an input sample is "0." Equations (3) and (4) are the formulas for calculating the FD and MFD, respectively:

$$D_i(x, y) = \sum_{n=1}^{N_i} (F_n(x, y)), \quad (3)$$

$$D_i(x, y) = \sum_{n=1}^{N_i} (F_n(x, y) * 2 - 1), \quad (4)$$

where x, y are coordinates of different pixels of any sample, $x = 1, \dots, k, y = 1, \dots, p, k, p$ are dimensions of normalized samples, n is the n th sample of a specified class, $F_n(x, y)$ is the pixel value of the n th sample at coordinate (x, y) , and N_i is the total number of samples in class C_i .

Figures 4 and 5 show examples of FD and MFD for digit "7" (digit "7") in the Arabic/Farsi digits set, obtained by using (3) and (4), respectively. For simplicity, only 200 samples of digit "7" were used for generating these figures. The reason for employing the MFD approach (4) to find similarity values in our research is that the MFD concept provides a more accurate description for similarity/distance between an image and a class template. Also, the MFD matrixes are considered as templates for different classes, called Template Matrixes (TMs).

3.2.2. Template Binarization. The generated Template Matrixes (TMs) include a considerable amount of information

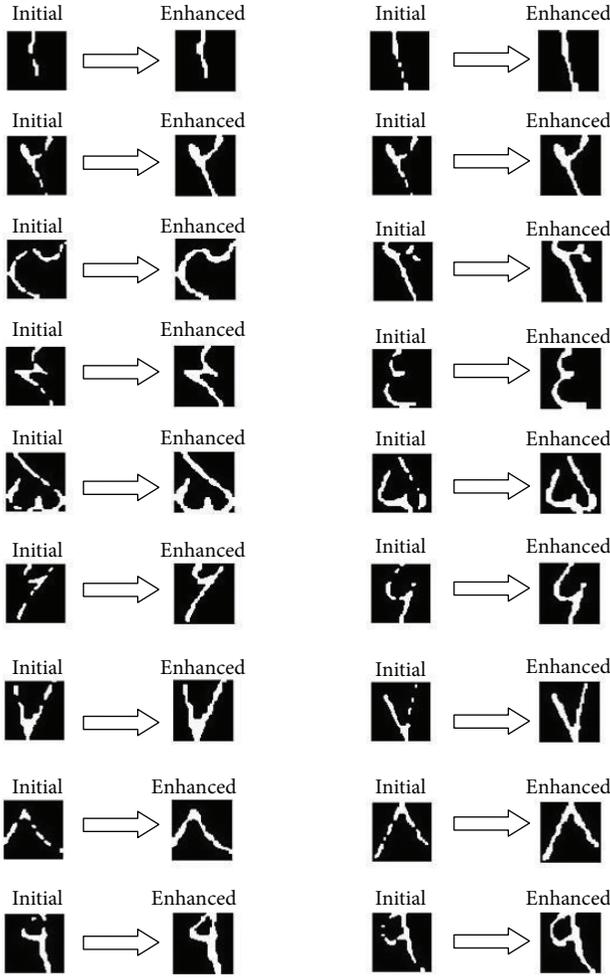


FIGURE 2: Applying preprocessing operations on Arabic/Farsi digits.

about the distribution of the pixels in each class. Hence, they can be considered to be a source for feature extraction in a PR system that uses the template matching technique.

The elements of a Template Matrix are computed by using MFD equation (4). Hence, they have a value between $-N_i$ and N_i . In order to compare the samples of a class with the derived template for the same class, the calculated numerical values obtained using (4) must be converted to binary numbers 0 and 1. In other words, it is necessary to convert the Template Matrices (TMs) to the Binarized Template Matrices (BTMs) version. This operation involves two steps. In the first step, we scaled the values of the TM elements to the gray levels spectrum from 0 to 255 by using (5), where N_i is the total number of samples in class C_i . In this equation, P is the initial value of each pixel in a TM:

$$\text{Gray_Level_TM} = \frac{(P + N_i)}{2} * \frac{255}{N_i}. \quad (5)$$

In the second step, the Gray_Level_TM elements are converted to the BTMs version by using the standard global Otsu's method. Figure 6 shows the BTM related to the

template obtained in Figure 5 using the above-mentioned method.

3.2.3. Computing Similarity Value. The use of template matching techniques involves determining similarities between an input instance and all generated class templates. There are various approaches for measuring the similarity between an input data and a class template, some of which include nonmetric cosine [35], conventional definition, modified frequency density [33], Hamming distance, linear correlation, cross correlation [36], Sawaki measure, and Rogers-Tanimoto measure. Various other definitions for similarity measures and distances that have been described in the literature include Minkowski distance (L_p distance), Bottleneck distance, Hausdorff distance, fringe distance, Turning Function distance, Fréchet distance, Reflection distance, and Transport distance (Earth Mover's distance) [37].

Using the Modified Frequency Diagram (4), we defined a similarity variable S . Similarity variable $S_{k,i}$ indicates the similarity value between the k th sample of class i and the corresponding class template. It is increased by the value of the MFD_i , if an image pixel and its corresponding template pixel have the same value of "1" or "0"; otherwise, $S_{k,i}$ is decreased. Also, in order to amplify the effect of similar pixels in comparison to nonsimilar pixels, we considered the effect of the equal pixels to be twice that of nonequal pixels, and we defined the general form of this new concept as follows (while the w reward coefficient is set to 2):

$$S_{k,i} = \sum_{x=1}^n \sum_{y=1}^m \{w * [f_{k,i}(x, y) \odot \text{BTM}_i(x, y)] - [f_{k,i}(x, y) \oplus \text{BTM}_i(x, y)]\} * |MFD_i(x, y)|, \quad (6)$$

where i is the class number in pattern space, n, m are image dimensions, w is the reward coefficient (in this research, this parameter was set to 2), $f_{k,i}$ is k th sample image of class i , BTM_i is i th Binarized Template Matrix (corresponding to class C_i), MFD_i is i th Modified Frequency Diagram matrix (corresponding to class C_i), \odot is logical XNOR operator, and \oplus is logical XOR operator.

Experimental results show that the calculated values for similarity variables $S_{k,i}$ proposed in (6) have wide variances, and this leads to better differentiation between the samples in a class. We set the w reward coefficient in (6) to (2). When w coefficient is increased, it will increase the effect of similar corresponding pixels in calculating the similarity values. It must be noted, however, that choosing too big a value for w will cancel the fine effect related to the corresponding nonsimilar pixels in (6).

3.2.4. Sieving Operation. The proposed similarity equation (6) was applied to the training dataset and a similarity value was computed for every sample in each class in order to increase overall system speed. Finally, all the training samples in a particular class were sorted in descending order based on their computed similarity values.

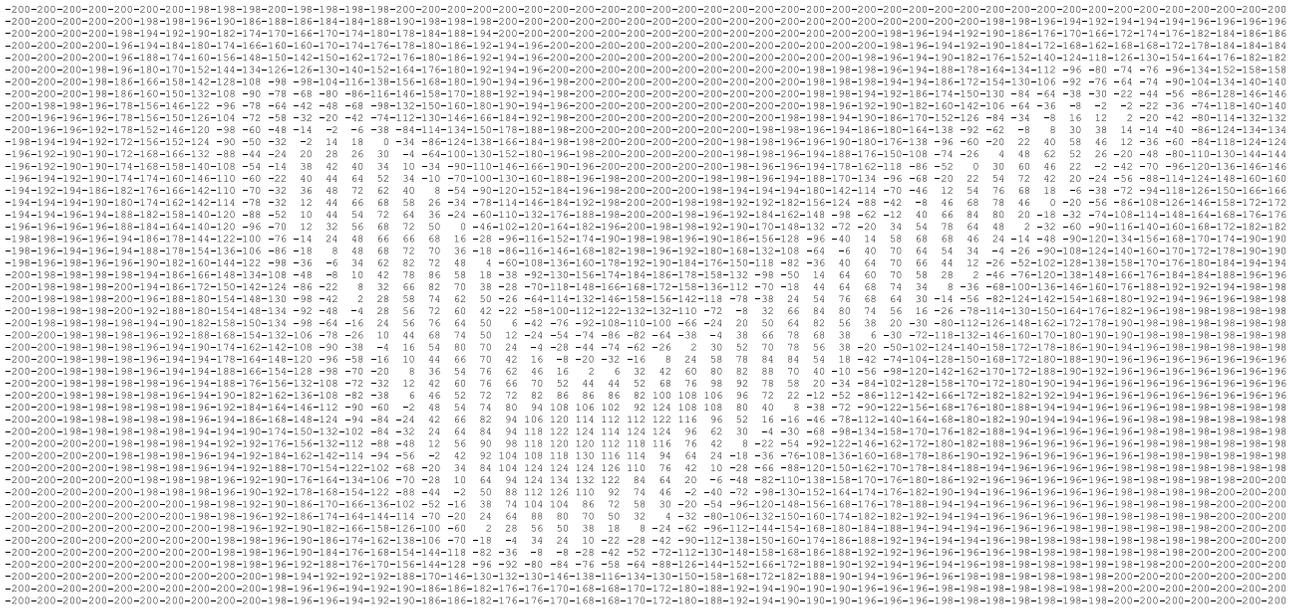


FIGURE 5: Modified Frequency Diagram (MFD) matrix for Arabic/Farsi digit “7” using (4).

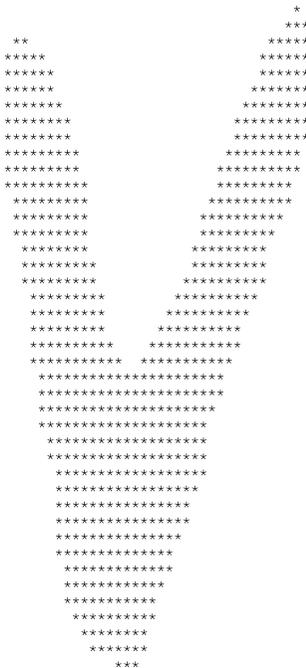


FIGURE 6: Binarized Template Matrix (BTM) corresponding to Figure 5.

4. Experimental Results

4.1. Hoda Dataset. All PR systems need initial datasets for training the system. In recent years, researchers have produced some benchmark datasets in order to encourage more researches in the PR domain and also to compare the functionalities of various PR systems that have been developed.

This research was conducted using the Arabic/Farsi OCR datasets and more specifically on handwritten Farsi/Arabic texts. Some available handwritten Farsi standard datasets with digits section include IFHCDB, Hoda, CENPARMI, and Hadaf [38]. Similar datasets for handwritten Arabic alphabets with digits part are Al-Isra, CENPARMI, ARABASE, and LMCA [39].

The digit part of the Hoda dataset [33]—one of the largest Farsi (and also Arabic) handwritten standard datasets—was chosen to test the proposed method. The Hoda dataset has two sections digits and characters. The digit section was prepared in 2007 by extracting the images of digits from 11,942 application forms for university entrance. Those forms were scanned at 200 dpi in 24-bit color format. The digits were extracted from the *postal code*, *national code*, *record number*, *identity certificate number*, and *phone number* fields of each form. The digit section of the Hoda dataset contains 80,000 samples and is divided into two parts—60,000 training samples and 20,000 testing samples. Figure 7 shows some sample digits from this dataset.

4.2. Proposed Method. In this research, the same operations were carried out in the preprocessing step on the training and testing samples. The outputs were noise-filtered, reslanted, relocated, and dimension-normalized. To save time and memory requirement, we rescaled all the preprocessed images into 20 × 20 pixel images again.

One template was created for each class by using all the training samples in the training part of the Hoda dataset and the Modified Frequency Diagram in (4). The templates were binarized using Otsu’s method. The PCA technique was applied on the training dataset to extract more important features automatically. For this task, all 400 pixels of any images were fed directly to the PCA algorithm. The data

0	1	2	3	4	5	6	7	8	9
۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
۰	۱	۲	۳	۴	۵	۶	۷	۸	۹

FIGURE 7: Some sample digits in the Hoda dataset.

was then mapped into the new orthogonal space based on the derived eigenvectors. Following that, we employed the proposed similarity function (6) and computed a similarity value for each training data sample. Based on these computed values, the training data in each class were sorted in descending order. Using sampling operation, the sorted dataset was sieved into 1/2, 1/3, and 1/4 of the original dataset volume for being used in further experiments.

It is very important to select a good classifier for a PR system. For this reason, some researchers use a combination of classifiers to achieve better results [40]. In our research, however, we employed a k -nearest neighbor (k -NN) classifier with Euclidean distance in the recognition phase to focus only on the power of the feature extraction block.

The k -NN is a simple and fast supervised machine-learning algorithm which is used to classify the unlabeled testing set with a labeled training set. In order to classify a new instance, the system finds the k -nearest neighbors among the training dataset to the new input sample and uses the categories of the k -nearest neighbors to weight the category candidates. The prediction class of the testing input is found based on the minimum distance between the testing input data and the training samples [19].

The k -NN algorithm can be described by

$$Y(d_i) = \operatorname{argmax}_{x_j \in k\text{-NN}} \operatorname{Sim}(d_i, x_j) y(x_j, c_k), \quad (7)$$

where d_i is the testing sample, x_j is one of the neighbors in the training set, $\operatorname{Sim}(d_i, x_j)$ is the similarity function for d_i , and $y(x_j, c_k) \in \{0, 1\}$ indicates whether x_j belong to class c_k .

Finally, the class with maximal sum of similarity will be selected as testing sample class. The similarity function can be Euclidian distance, Mahalanobis distance, fringe distance, and so on.

Figure 8 shows the system's accuracy versus the number of features which are selected from an initial feature vector by the PCA. In general, accuracy peaks at a certain interval of features and then diminishes or saturates. In this experiment, the highest accuracies were achieved at intervals of 40 to 100 features.

To find the optimum number of features, the experiment was repeated with different number of features at intervals of

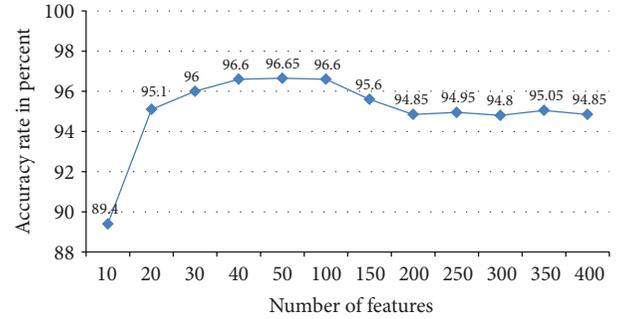


FIGURE 8: Accuracy versus the number of features proposed by PCA.

30 to 105 features using step 5. The best results were obtained for interval [75, 85]. The experiment was repeated for interval [75, 85] with step 1. Finally, the highest accuracy of 97.11% was achieved by using the first 79 features of the feature vector.

In the second experiment, and based on the results shown in Figure 8, the first 79 features proposed by PCA were selected as the final feature vector. In order to increase the recognition speed and also to evaluate the performance of the proposed sieving operation, we used different versions of the sieved training dataset. In all experiments, the number of features was 79. The results are shown in Table 1. This table clearly shows that the recognition speed increases to more than double, while the accuracy decreases slightly from 97.11% to 96.39%. If the reduced 1/3 version of the training dataset is used, the recognition speed will be more than 3 times faster, but the accuracy will drop by 3.03% from 97.11% to 94.08%.

4.3. Results Comparison. To compare the results in OCR domain, the results produced by the use of the proposed approach were compared with that from other available methods reported in the literature. Table 2 shows these comparisons. It is obvious that in most of the previous researches, MLP-NN was employed as the recognition engine. In order to have a better comparison, we repeated our experiments but used MLP-NN instead of k -NN at the recognition stage. At this stage, an MLP-NN was trained using 79 neurons in the input layer (corresponding to the number of selected features as in the previous experiment), 30 neurons in the hidden layer (found experimentally), and 10 neurons in the output layer (corresponding to 10 different classes of digits from 0 to 9), respectively. Each experiment was repeated 10 times and the obtained results have been reported in average. The achieved results have been shown in the last few rows in Table 2.

The results from the proposed method are better than the results for the other researches mentioned in Table 2, in terms of accuracy, except for the result obtained by Ziaratban et al. [31]. However, an accurate comparison between the proposed approach and other researches is not possible, because of the different databases used, number of training and testing samples, number of features, and the classifier employed.

To have a more accurate comparison with the recent related researches in Farsi OCR domain, we repeated our proposed approach under the same conditions in the method used by Enayatifar and Alirezanejad [32]. Both researches

TABLE 1: Recognition rate for different training dataset volumes.

Number of samples used for recognition operation by k -NN	Recognition time for a sample (seconds)	Ratio of recognition time to initial recognition time $T1$ (in percent)	Accuracy		
			$K = 1$	$K = 3$	$K = 5$
60,000	$T1 = 0.1161352$	$T1/T1 = 100\%$	97.11%	95.93%	95.70%
30,000	$T2 = 0.0509572$	$T2/T1 = 43.87748\%$	96.39%	94.97%	94.91%
20,000	$T3 = 0.0367815$	$T3/T1 = 31.67128\%$	94.08%	93.66%	93.61%
15,000	$T4 = 0.0296006$	$T4/T1 = 25.48805\%$	93.57%	93.40%	93.17%

TABLE 2: Result comparison for handwritten Farsi/Arabic digit recognition.

Year	References	Training dataset	Total number of testing samples	Number of features	Classifier	Accuracy
2003	Mowlaei and Faez [27]	(Private) (only 8 digits)	1600	64	SVM	92.44%
2003	Sadri et al. [28]	CENPARMI ver. 1	3035	64	MLP-NN	91.25%
2004	Mozaffari et al. [29]	(Private) (only 8 digits)	1600	64	MLP-NN	91.37%
2005	Mozaffari et al. [25]	(Private) (only 8 digits)	1600	240	Fr. NN	86.30%
2007	Ziaratban et al. [31]	(Private)	4000	60	MLP-NN	97.65%
2011	Enayatifar and Alirezanejad [32]	Hoda (60,000 samples)	20,000	48	MLP-NN	92.70%
2013	The proposed method	Hoda (60,000 samples)	20,000	79	k -NN	97.11%
2013	The proposed method	Hoda (30,000 samples)	20,000	79	k -NN	96.39%
2013	The proposed method	Hoda (20,000 samples)	20,000	79	k -NN	94.08%
2013	The proposed method	Hoda (15,000 samples)	20,000	79	k -NN	93.57%
2013	The proposed method	Hoda (60,000 samples)	20,000	79	MLP-NN	96.27%
2013	The proposed method	Hoda (30,000 samples)	20,000	79	MLP-NN	95.14%
2013	The proposed method	Hoda (20,000 samples)	20,000	79	MLP-NN	90.71%
2013	The proposed method	Hoda (10,000 samples)	20,000	79	MLP-NN	82.86%

have employed the same dataset, equal number of testing samples, and the same classifier (MLP-NN). However, the number of features in their study is 48, but we made a 79-dimensional feature vector. It is obvious that this value cannot decrease to 48. Also, our proposed approach introduces a new method for dataset reduction. Table 3 compares our proposed method results to the method proposed by Enayatifar and Alirezanejad in terms of accuracy and speed. Our proposed approach not only outperforms by achieving higher accuracy, but also it is about 4.5 times faster than the latter method.

4.4. Error Analysis. One of the main problems in Arabic/Farsi digit recognition using the template matching technique is that there is more than one general shape for digits 2, 3, 4, 5, and 6. Each of these digits is usually written in two (or more) different shapes in handwritten documents. Hence, the number of classes for Arabic/Farsi digits is 16 and not 10. These extra patterns degrade the templates generated for these digits. Hence, some templates are not similar enough to the sample images, and this causes the recognition system to produce the wrong results. To overcome this drawback, more than one template for the stated digits should be considered. Figure 9 shows the various shapes of the Arabic/Farsi digits 2, 3, 4, 5, and 6.

Another main source of errors in this research is the degraded samples in the testing part of the Hoda dataset, which cannot even be enhanced by the preprocessing block.

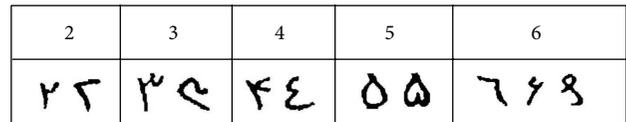


FIGURE 9: Different shapes for some of Arabic/Farsi digits.

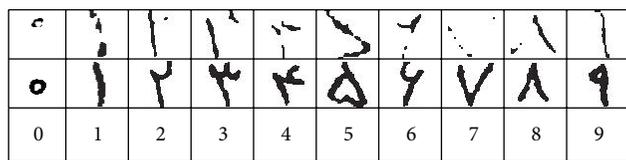


FIGURE 10: Some degraded digit samples (first row) in the Hoda dataset.

It is also worth noting that there are usually some completely degraded samples as well as wrong samples in standard datasets (taken from real data) that are used to investigate the ability and behavior of a PR system to deal with these wrong or outlier samples. The Hoda dataset follows this rule. Figure 10 shows some degraded samples in the digit part of the Hoda dataset. Figure 11 shows the images of some degraded samples for digit 4 that had been misclassified as digits 2 or 3. These highly degraded samples have a negative impact not only on the template generating process but also on the recognition process.

TABLE 3: Result comparison for handwritten Farsi/Arabic digit recognition.

Year	References	Training dataset	Total number of testing samples	Number of features	Recognition time per sample (seconds)	Accuracy
2013	The proposed method	Hoda (40,000 samples)	20,000	79	0.036810	96.02%
2011	Enayatifar and Alirezanejad [32]	Hoda (40,000 samples)	20,000	48	(Not reported)	92.70%
2013	The proposed method	Hoda (7,000 samples)	3000	79	0.046235	95.72%
2011	Enayatifar and Alirezanejad [32]	Hoda (7,000 samples)	3000	48	0.21	94.30%

TABLE 4: Confusion matrix for the proposed method.

	۰ (0)	۱ (1)	۲ (2)	۳ (3)	۴ (4)	۵ (5)	۶ (6)	۷ (7)	۸ (8)	۹ (9)
۰ (0)	2000	0	0	0	0	0	0	0	0	0
۱ (1)	0	1998	2	0	0	0	0	0	0	0
۲ (2)	0	21	1928	36	5	2	2	6	0	0
۳ (3)	0	0	133	1823	37	2	0	3	2	0
۴ (4)	3	8	61	56	1856	2	7	3	0	4
۵ (5)	0	7	0	0	3	1966	3	2	14	5
۶ (6)	2	6	10	2	3	0	1952	5	2	18
۷ (7)	0	2	9	2	2	0	4	1979	0	2
۸ (8)	0	4	0	0	0	0	3	0	1987	6
۹ (9)	0	18	6	0	0	3	26	0	4	1943



FIGURE 11: Some samples of digit 4 which were misclassified as digit 2 or digit 3.

As shown in the confusion matrix in Table 4, more than 57.74% of the errors are related to the misclassification of very similar digits 2, 3, and 4, in the handwriting mode. Most of the OCR systems for Arabic/Farsi language suffer from this too similar characteristic.

5. Conclusion

In this paper, a new dataset size reduction method was proposed in order to overcome the time complexity problem and to speed up the training and testing operations in an OCR application. To achieve this goal, we created a Modified Frequency Diagram, and we also developed a new method for creating a template for each class in the pattern space, while the similarity between each sample and its corresponding template was computed. This new similarity function was used to sort all the training data in each class based on the similarity to class template. After sorting all the data in each class, the training dataset size was sieved to 1/2, 1/3, and 1/4 of the original size by sampling at a rate of 1/2, 1/3, and 1/4.

In order to reduce the dimensionality, a PCA-based approach was introduced for automatic feature extraction and also for features selection in an OCR application for handwritten texts. Features were first extracted from the training patterns automatically by using the standard PCA. Then, using m biggest eigenvalues and their corresponding eigenvectors, a suboptimal feature set was selected from the initial feature vector. The input patterns were then mapped into the new orthogonal pattern space. A smaller dataset which has smaller number of samples and also smaller number of features was produced.

In the recognition stage, any input instance is mapped to a new pattern space using the computed suboptimal feature vector in the training phase. Classification is carried for an input instance by finding the nearest neighbor in the sieved dataset. The experiment was repeated using the MLP-NN instead of k -NN as the classification engine to compare the results with those reported in the literature.

The algorithm mentioned was implemented in an OCR system for recognizing digits in the Hoda dataset, which is one of the biggest standard handwritten Arabic/Farsi datasets. We achieved 97.11%, 96.39%, 94.08%, and 93.57% accuracy when all, 1/2, 1/3, and 1/4 training datasets, respectively, were used in the recognition phase using the k -NN classifier. In this case, the accuracy decreases from 97.11% to 96.39% while recognition speed increases by nearly two times. When a MLP-NN classifier was employed, the accuracies were 96.27%, 95.14%, 90.71%, and 82.86% corresponding to all, 1/2, 1/3, and 1/4 of training datasets, respectively. In

this case, the accuracy had decreased from 96.27% to 95.14%, while the recognition speed had increased by two times. Compared to one of the recent researches [32] in this domain, our proposed method achieved higher performance with recognition speed being about 4.5 times faster.

Identifying a systematic way to increase system recognition speed and introducing a new approach for dataset size reduction are two applications of the proposed method. The dataset size reduction algorithm is not only effective in OCR application—a subcategory of PR systems—but also can be used for dataset size reduction in other PR systems with different types of pictorial databases.

The current approaches used for dataset size reduction usually remove two groups of samples from the classes: near to classes' centroids and far from classes' centroids (outlier samples or support vector samples). However, the proposed method keeps some samples near to the classes' centers to make the system model better and also keeps some samples far from each class center to be able to assess system performance more accurately.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 99, pp. 1–24, 2013.
- A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for EMG signal classification," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7420–7431, 2012.
- H. B. Borges and J. C. Nievola, "Comparing the dimensionality reduction methods in gene expression databases," *Expert Systems with Applications*, vol. 39, pp. 10780–10795, 2012.
- M. Song, H. Yang, S. H. Siadat, and M. Pechenizkiy, "A comparative study of dimensionality reduction techniques to enhance trace clustering performances," *Expert Systems With Applications*, vol. 40, pp. 3722–3737, 2013.
- C. Yang, W. Zhang, J. Zou, S. Hu, and J. Qiu, "Feature selection in decision systems: a mean-variance approach," *Mathematical Problems in Engineering*, vol. 2013, Article ID 268063, 8 pages, 2013.
- G. A. Abandah, K. S. Younis, and M. Z. Khedher, "Handwritten Arabic character recognition using multiple classifiers based on letter form," in *Proceedings of the 5th IASTED International Conference on Signal Processing, Pattern Recognition & Applications (SPPRA '08)*, pp. 128–133, February 2008.
- W. Zhongdong, Y. Jianping, X. Weixin, and G. Xinbo, "Reduction of training datasets via fuzzy entropy for support vector machines," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 2381–2385, October 2004.
- S. V. N. Vishwanathan and M. N. Murty, "Use of multi category proximal SVM for data set reduction," *International Journal of Studies in Fuzziness and Soft Computing*, vol. 140, pp. 3–20, 2004.
- K. Hara and K. Nakayama, "Training data selection method for generalization by multilayer neural networks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E81-A, no. 3, pp. 374–381, 1998.
- C. Ding and X. He, "K-means clustering via principal component analysis," in *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, pp. 1–9, July 2004.
- S. Abdul Sattar and S. Shahl, "Character recognition of arabic script languages," in *Proceedings of the 2nd International Conference on Communication and Information Technology*, pp. 502–506, 2012.
- M. Elzobi, A. Al-Hamadi, L. Dinges, and B. Michaelis, "A structural features based segmentation for off-line handwritten Arabic text," in *Proceedings of the 5th International Symposium on I/V Communications and Mobile Networks (ISIVC '10)*, pp. 1–4, October 2010.
- H.-C. Kim, D. Kim, and S. Yang Bang, "A numeral character recognition using the PCA mixture model," *Pattern Recognition Letters*, vol. 23, no. 1-3, pp. 103–111, 2002.
- P. Zhang, C. Y. Suen, and T. D. Bui, "Multi-modal nonlinear feature reduction for the recognition of handwritten numerals," in *Proceedings of the 1st Canadian Conference on Computer and Robot Vision*, pp. 393–400, May 2004.
- V. Deepu, S. Madhvanath, and A. G. Ramakrishnan, "Principal component analysis for online handwritten character recognition," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 327–330, August 2004.
- A. Sharma and K. K. Paliwal, "Fast principal component analysis using fixed-point algorithm," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1151–1155, 2007.
- M. T. Parvez and S. A. Mahmoudi, "Offline Arabic handwritten text recognition : a survey," *ACM Computing Survey*, vol. 45, no. 2, article 23, 2013.
- R. Azmi, B. Pishgoo, N. Norozi, M. Koohzadi, and F. Baesi, "A hybrid GA and SA algorithms for feature selection in recognition of hand-printed Farsi characters," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS '10)*, vol. 3, pp. 384–387, October 2010.
- Y. El-Glaly and F. Quek, "Isolated Handwritten Arabic Character Recognition using Multilayer Perceptron and K Nearest Neighbor Classifiers," 2012, http://filebox.vt.edu/users/yasmineg/index.htm_files/MLSP%20Arabic%20Character%20Recognition.pdf.
- A. R. Kheyrkhah and E. Rahmadian, "Optimizing a Farsi handwritten character recognition system by selecting effective features on classifier using genetic algorithm," in *Proceedings of the 1st Joint Congress on Fuzzy and Intelligent Systems*, 2007 (Persian).
- A. M. Urmanov, A. A. Bougaev, and K. C. Gross, "Reducing the size of a training set for classification," US Patent no. 7478075 B2, 2007.
- I. Javed, M. N. Ayyaz, and W. Mehmood, "Efficient training data reduction for SVM based handwritten digits recognition," in *Proceedings of the International Conference on Electrical Engineering (ICEE '07)*, pp. 1–4, April 2007.
- B. Boucheham, "PLA data reduction for speeding up time series comparison," *International Arab Journal of Information Technology*, vol. 9, no. 5, 2012.
- J. Cervantes, X. Li, and W. Yu, "Support vector classification for large data sets by reducing training data with change of classes," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '08)*, pp. 2609–2614, October 2008.

- [25] S. Mozaffari, K. Faez, and M. Ziaratban, "Character representation and recognition using quadtree-based fractal encoding scheme," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 819–823, September 2005.
- [26] M. Ziaratban, K. Faez, and M. Ezoji, "Use of legal amount to confirm or correct the courtesy amount on Farsi bank checks," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07)*, pp. 1123–1127, September 2007.
- [27] A. Mowlaei and K. Faez, "Recognition of isolated handwritten Persian/Arabic characters and numerals using support vector machines," in *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing*, pp. 547–554, 2003.
- [28] J. Sadri, C. Y. Suen, and T. D. Bui, "Application of support vector machines of handwritten Arabic/Persian digits," in *Proceedings of the 2nd Iranian Conference on Machine Vision, Image Processing & Applications (MVIP '03)*, vol. 1, pp. 300–307, 2003.
- [29] S. Mozaffari, K. Faez, and H. R. Kanan, "Feature comparison between fractal codes and wavelet transform in handwritten alphanumeric recognition using SVM classifier," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 331–334, August 2004.
- [30] H. Soltanzadeh and M. Rahmati, "Recognition of Persian handwritten digits using image profiles of multiple orientations," *Pattern Recognition Letters*, vol. 25, no. 14, pp. 1569–1576, 2004.
- [31] M. Ziaratban, K. Faez, and F. Faradji, "Language-based feature extraction using template-matching in Farsi/Arabic handwritten numeral recognition," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07)*, pp. 297–301, September 2007.
- [32] R. Enayatifar and M. Alirezanejad, "Offline handwriting digit recognition by using direction and accumulation of pixels," in *Proceedings of the International Conference on Computer and Software Modeling*, vol. 14, pp. 214–220, 2011.
- [33] H. Khosravi and E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on their varieties," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1133–1141, 2007.
- [34] M. Hanmandlu, K. R. Murali Mohan, S. Chakraborty, S. Goyal, and D. R. Choudhury, "Unconstrained handwritten character recognition based on fuzzy logic," *Pattern Recognition*, vol. 36, no. 3, pp. 603–623, 2003.
- [35] A. C. Downton, E. Kabir, and D. Guillevic, "Syntactic and contextual post processing of handwritten addresses for optical character recognition," in *Proceedings of the 9th International Conference on Pattern Recognition*, pp. 1072–1076, 1988.
- [36] T. Sitamahalakshmi, A. Vinay Babu, and M. Jagadeesh, "Character recognition using Dempster-Shafer theory—combining different distance measurement methods," *International Journal of Engineering and Technology*, vol. 2, no. 5, pp. 1177–1184, 2010.
- [37] V. Curic, J. Lindblad, N. Sladoje, H. Sarve, and G. Borgfors, "A new set distance and its application to shape registration," *Pattern Analysis and Applications*, vol. 12, pp. 1–12, 2012.
- [38] A. Alaei, P. Nagabhushan, and U. Pal, "A new dataset of Persian handwritten documents and its segmentation," in *Proceedings of the 7th Iranian Conference on Machine Vision and Image Processing (MVIP '11)*, pp. 1–5, November 2011.
- [39] M. Kherallah, A. Elbaati, H. E. Abed, and M. Alimi, "The on/off (LMCA) dual arabic handwriting database," in *Proceedings of the International Conference on Frontiers in Handwriting Recognition*, 2008.
- [40] Y.-C. Chim, A. A. Kassim, and Y. Ibrahim, "Dual classifier system for handprinted alphanumeric character recognition," *Pattern Analysis and Applications*, vol. 1, no. 3, pp. 155–162, 1998.

Research Article

A Global Multilevel Thresholding Using Differential Evolution Approach

Kanjana Charansiriphaisan, Sirapat Chiewchanwattana, and Khamron Sunat

Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

Correspondence should be addressed to Khamron Sunat; khamron_sunat@yahoo.com

Received 3 October 2013; Revised 23 January 2014; Accepted 3 February 2014; Published 20 March 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Kanjana Charansiriphaisan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Otsu's function measures the properness of threshold values in multilevel image thresholding. Optimal threshold values are necessary for some applications and a global search algorithm is required. Differential evolution (DE) is an algorithm that has been used successfully for solving this problem. Because the difficulty of a problem grows exponentially when the number of thresholds increases, the ordinary DE fails when the number of thresholds is greater than 12. An improved DE, using a new mutation strategy, is proposed to overcome this problem. Experiments were conducted on 20 real images and the number of thresholds varied from 2 to 16. Existing global optimization algorithms were compared with the proposed algorithms, that is, DE, rank-DE, artificial bee colony (ABC), particle swarm optimization (PSO), DPSO, and FODPSO. The experimental results show that the proposed algorithm not only achieves a more successful rate but also yields a lower threshold value distortion than its competitors in the search for optimal threshold values, especially when the number of thresholds is large.

1. Introduction

Thresholding is the simplest and most commonly used method of image segmentation. It can be bilevel or multilevel [1]. Both of these types can be classified into parametric and nonparametric approaches [1]. Surveys of thresholding techniques for image segmentation can be found in [2–7]. The surveys revealed that Otsu's method is a commonly used technique [4, 8]. This method finds the optimal thresholds by maximizing the weighted sum of between-class variances (BCV) [9]. The BCV function is also called Otsu's function. However, the solution finding process is an exhaustive search and it is a very time-consuming process because the complexity grows exponentially with the number of thresholds.

Multilevel image thresholding based on Otsu's function has been used as a benchmark for comparing the capability of evolutionary algorithms (EA). The EA is a nongradient based optimization algorithm. Several algorithms have been widely applied to solve multilevel thresholding. A group of successful works were based on a combination of Otsu's function with some state-of-the-art algorithms: PSO [10], DE [11], ABC [12], and FOSPSO [13]. Kulkarni and Venayagamoorthy [14]

showed that PSO was faster than Otsu's method in searching the optimal thresholds of multilevel image thresholding. Akay [15] presented a comprehensive comparative study of the ABC and PSO algorithms. The results showed that the ABC algorithm with both the between-class variance and the entropy criterion can be efficiently used in multilevel thresholding. Hammouche et al. [16] focused on solving the image thresholding problem by combining Otsu's function with metaheuristic techniques, that is, genetic algorithm (GA), PSO, DE, ant colony, simulated annealing, and Tabu search. Their results revealed that DE was the most efficient with respect to the quality of solution. Osuna-Enciso et al. [17] presented an empirical comparative study of the ABC, PSO, and DE algorithms to perform image thresholding using a mixture of Gaussian functions. The results showed that the DE algorithm was superior in performance in minimizing the Hellinger distance and used less evaluations of the Hellinger distance. Ghamisia et al. [18] showed that a global optimal search for optimal threshold values of Otsu's function was essential for the multilevel segmentation of multispectral and hyperspectral images.

The DE algorithm was selected for multilevel image thresholding. It is simple to implement and produces good results. However, based on our experiments, DE could not reach an optimal solution when it was applied to a very difficult problem. Therefore, a better DE algorithm is required. We noticed that the mechanism of vector selection and the size of the higher ranked population are an important criterion for success.

The contribution of this paper is as follows.

DE with the onlooker and ranking-based mutation operation, named $O(\beta)R$ -DE, is proposed to overcome the drawback of the DE algorithm for multilevel image thresholding, especially when the number of thresholds is large. The proposed algorithm homogenizes the onlooker phase of the ABC algorithm and the ranking-based mutation operator of the rank-DE [19]. The main advantage of the proposed algorithm is that a user can adjust the balancing of the exploitation and exploration capabilities of the algorithm.

To verify the capabilities of the proposed $O(\beta)R$ -DE algorithm, experiments to find the optimal solutions in the multilevel image thresholding, when the number of thresholds ranged from two to 16, were set up. It was found that the optimal solutions could be effectively reached using the proposed $O(\beta)R$ -DE algorithm.

The remainder of the paper is organized as follows. Section 2 describes the multilevel thresholding problem. Section 3 presents a brief review of the differential evolution algorithm (DE). In Section 4, the proposed new version of the DE algorithm with the onlooker and ranking-based mutation operator algorithm, $O(\beta)R$ -DE, is described in detail. Section 5 shows the experimental results of applying the proposed method to multilevel segmentation in different images. Finally, the conclusion of the paper is discussed in Section 6.

2. Multilevel Thresholding Problem Formulation

Otsu's method [9] is based on the maximization of the between-class variance. Consider a digital image having the size $H \times W$, where W is the width and H is the height. The pixels of a given picture are represented in L gray levels and they are in $\{0, 1, 2, \dots, L - 1\}$. The number of pixels at level i is denoted by n_i and the total number of pixels by $N = n_1 + n_2 + \dots + n_L$. The gray-level histogram is normalized and regarded as a probability distribution and is written as follows:

$$p_i = \frac{n_i}{N}, \quad p_i \geq 0, \quad \sum_{i=1}^L p_i = 1. \quad (1)$$

The total mean of the image can be defined as

$$\mu_T = \sum_{i=1}^L i \times p_i. \quad (2)$$

The multilevel thresholding with respect to the given $n - 1$ threshold values t_j , $j = 1, \dots, n - 1$ can be performed as follows:

$$F(x, y) = \begin{cases} 0, & f(x, y) \leq t_1, \\ \frac{1}{2}(t_1 + t_2), & t_1 < f(x, y) \leq t_2, \\ \vdots & \\ \frac{1}{2}(t_{n-2} + t_{n-1}), & t_{n-2} < f(x, y) \leq t_{n-1}, \\ L, & f(x, y) > t_{n-1}, \end{cases} \quad (3)$$

where (x, y) is the coordinate of a pixel and $f(x, y)$ denotes the intensity level of a pixel. The pixels of a given image will be divided into n classes D_1, \dots, D_n in this regard.

The optimal threshold can be determined by maximizing the between-class variance function (BCV), σ_B^2 , which can be defined by

$$\sigma_B^2 = \sum_{j=1}^n w_j (\mu_j - \mu_T)^2, \quad (4)$$

where j represents a specific class in such a way that w_j and μ_j are the probability of occurrence and the mean of class j , respectively. Equation (4) is also called Otsu's function. The probabilities of occurrence w_j of classes D_1, \dots, D_n are defined by

$$w_j = \begin{cases} \sum_{i=1}^{t_j} p_i, & j = 1, \\ \sum_{i=t_{j-1}+1}^{t_j} p_i, & 1 < j < n, \\ \sum_{i=t_{j-1}+1}^L p_i, & j = n. \end{cases} \quad (5)$$

The mean of each class μ_j can be given by

$$\mu_j = \begin{cases} \sum_{i=1}^{t_j} \frac{i \times p_i}{w_j}, & j = 1, \\ \sum_{i=t_{j-1}+1}^{t_j} \frac{i \times p_i}{w_j}, & 1 < j < n, \\ \sum_{i=t_{j-1}+1}^L \frac{i \times p_i}{w_j}, & j = n. \end{cases} \quad (6)$$

Thus, the n -level thresholding problem is transformed to an optimization problem. The process is to search for $n - 1$ thresholds t_j that maximize the value φ , which is generally defined as

$$\varphi = \max_{1 < t_1 < \dots < t_{n-1} < L} \sigma_B^2(t_j). \quad (7)$$

3. Differential Evolution Algorithm

The DE algorithm is an evolutionary optimization technique proposed by Storn and Price [11]. The main procedures of DE are briefly described as follows.

3.1. Initialization. The DE algorithm starts with a population of initial solutions, each of dimension D , $X_{i,g} = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$, $i = 1, \dots, \text{NP}$, where the index i denotes the i th solution, or vector, of the population, g is the generation, and NP is the population size. The initial population (at $g = 0$) is randomly generated to be within the search space constrained by the minimum and maximum bounds, $X_{\min} = \{x_{1,\min}, x_{2,\min}, \dots, x_{D,\min}\}$ and $X_{\max} = \{x_{1,\max}, x_{2,\max}, \dots, x_{D,\max}\}$. The i th vector x_i is initialized as follows:

$$x_{j,i,0} = x_{j,\min} + \text{rndreal}_{i,j}[0, 1] \cdot (x_{j,\max} - x_{j,\min}), \quad (8)$$

where $\text{rndreal}_{i,j}[0, 1]$ is a uniformly distributed random real number between 0 and 1, ($0 \leq \text{rndreal}_{i,j}[0, 1] < 1$).

3.2. Mutation Operators. The differential mutation operator is one of the three operators of DE. The mutation operator is applied to generate the mutant vector v_i for each target vector x_i in the current population. A mutant vector is generated according to

$$v_{i,g+1} = x_{r_1,g} + F \cdot (x_{r_2,g} - x_{r_3,g}), \quad (9)$$

where the randomly chosen indexes, random indexes, $r_1, r_2, r_3 \in \{1, 2, \dots, \text{NP}\}$ are mutually different random integer indices and they are also different from the running index i . Further, i, r_1, r_2 , and r_3 are different so that $\text{NP} \geq 4$. F is a real and constant factor, $F \in [0, 2]$, which controls the amplification of the differential variation; $x_{r_1,g}$ is called the base vector, $x_{r_2,g}$ is called the terminal vector, $x_{r_3,g}$ is called the other vector, and $(x_{r_2,g} - x_{r_3,g})$ is called the difference vector.

There have been many proposed mutation strategies for DE [20, 21]. Each different strategy has different characteristics and is suitable for a set of problems. However, the choice of the best mutation operators for DE is difficult for a specific problem [22–24]. The “DE/rand/1/bin” strategy has been widely used in DE literature [25–28]. It is more reliable than the strategies based on the best-so-far solution such as “DE/best/1” and “DE/current-to-best/1”. However, “DE/rand/1/bin” has slower convergence. Simply put, it has high exploration but low exploitation abilities.

3.3. Crossover. DE utilizes the crossover operation to generate new solutions by shuffling competing vectors and to increase the diversity of the population. The classical version of the DE (DE/rand/1/bin) uses the binary crossover. It defines the following trial vector:

$$u_{i,g+1} = (u_{1i,g+1}, u_{2i,g+1}, \dots, u_{Di,g+1}), \quad (10)$$

where $j = 1, \dots, D$ (D = problem dimension) and

$$u_{ji,g+1} = \begin{cases} v_{ji,g+1} & \text{if } (\text{randb}(j) \leq \text{CR}) \text{ and } j = \text{rnbr}(i) \\ x_{ji,g} & \text{if } (\text{randb}(j) > \text{CR}) \text{ and } j \neq \text{rnbr}(i). \end{cases} \quad (11)$$

CR is the crossover rate $\in [0, 1]$, $\text{randb}(j)$ is the j th evaluation of a uniform random number generator with outcome $\in [0, 1]$, and $\text{rnbr}(i)$ is a randomly chosen index $\in 1, 2, \dots, D$ that ensures $u_{i,g+1}$ will get at least one parameter from $v_{i,g+1}$.

3.4. Selection. Selection determines whether the target or the trial vector survives to the next generation. The selection operation is described as

$$x_{i,g+1} = \begin{cases} u_{i,g}, & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g}, & \text{if } f(u_{i,g}) > f(x_{i,g}), \end{cases} \quad (12)$$

where $f(x)$ is the objective function to be minimized. Therefore, if the objective of the new trial vector, $f(u_{i,g})$, is equal to or less than the objective of the old trial vector, $f(x_{i,g})$, then $x_{i,g+1}$ is set to $u_{i,g}$; otherwise, the old value $x_{i,g}$ is retained.

The pseudocode of basic DE with “DE/rand/1/bin” strategy is shown in Algorithm 1.

The function $\text{rndint}[1, D]$ returns a uniformly distributed random integer number between 1 and D . $\text{rndreal}_j[0, 1]$ is a uniformly distributed random real value of $[0, 1]$. The word “better” in line 17 means “less than” if the problem requires minimization, see (12) and its explanation, and it means “greater than,” if the problem requires maximization. The best $X_{i,G}$, where G is the maximum number of generations, is the solution of the algorithm. The word “best” also depends on the type of problem.

4. The Proposed DE with Onlooker Ranking-Based Mutation Operator

In 2013 Gong and Cai [19] proposed a rank-DE algorithm. They claimed that probabilistically selecting the vectors x_{r_1} and x_{r_2} in the mutation operator from the better population can improve the exploitation ability of basic DE. To the best of the authors’ knowledge, rank-DE may, however, also lead to premature convergence (this will be shown in the experiments). That means that the rank-DE has too much exploitation ability. Furthermore, it cannot balance between the exploration and the exploitation abilities. In order to balance between the two abilities, we propose DE with the onlooker and ranking-based mutation operator, named $O(\beta)R$ -DE. The proposed algorithm is an improvement of the rank-DE by homogenizing the rank-DE with the onlooker phase of ABC algorithm. The detail of the $O(\beta)R$ -DE algorithm is described as follows.

4.1. Ranking Assignment. To perform the maximization, the fitness of each vector is sorted in ascending order (i.e., from worst to best). Then, the rank of the i th vector, R_i , is assigned based on its sorted ordering as follows:

$$R_{\text{order}} = \text{order}, \quad \text{order} = 1, 2, \dots, \text{NP}. \quad (13)$$

As a result, the best vector in the current population will obtain the highest ranking, that is, NP.

```

(1) Generate the initial population randomly
(2) Evaluate the fitness for each individual in the population
(3) while the maximum generation  $G$  is not reached do
(4)   for  $i = 1$  to NP do
(5)     Select uniform randomly  $r_1 \neq r_2 \neq r_3 \neq i$ 
(6)      $j_{\text{rand}} = \text{rndint}[1, D]$ 
(7)     for  $j = 1$  to  $D$  do
(8)       if  $\text{rndreal}_j[0, 1] \leq \text{CR}$  or  $j$  is equal to  $j_{\text{rand}}$  then
(9)          $u_{i,j} = x_{r_1,j} + F \cdot (x_{r_2,j} - x_{r_3,j})$ 
(10)        else
(11)           $u_{i,j} = x_{i,j}$ 
(12)        end if
(13)      end for
(14)    end for
(15)  for  $i = 1$  to NP do
(16)    Evaluate the offspring  $u_i$ 
(17)    if  $f(u_i)$  is better than or equal to  $f(x_i)$  then
(18)      Replace  $x_i$  with  $u_i$ 
(19)    end if
(20)  end for
(21) end while

```

ALGORITHM 1: The DE algorithm with “DE/rand/1/bin” strategy.

4.2. *Probabilistic Selection.* After assigning the ranking for each vector, the selection probability p_i of the i th vector x_i is calculated as

$$p_i = \frac{R_i}{\text{NP}}, \quad i = 1, 2, \dots, \text{NP}. \quad (14)$$

4.3. A New Strategy for Base Vector, Terminal Point, and the Other Vector Selections

Definition 1 (a worse population and a better population). Let ζ be a real value and $0 \leq \zeta < 1$. A population having probability less than ζ is called a worse population and a population having probability greater than or equal to ζ is called a better population.

In the rank-DE, the base vector x_{r_1} and the terminal point x_{r_2} were based on their selection probabilities. The other vector in the mutation operator, x_{r_3} , is selected randomly as in the original DE algorithm. The vectors with higher rankings (higher selection probabilities) are more likely to be chosen as the base vector or the terminal point in the mutation operator.

Our investigation revealed that if both x_{r_1} and x_{r_2} vectors of rank-DE were chosen from better vectors, then the distribution of the target vector may collapse quickly and possibly lead to premature convergence. Accordingly, when the rank-DE was applied to a very difficult problem, it could not reach the optimal solution.

If the steps of the DE algorithm are compared with the ABC algorithm, the population in the current generation can be considered as the employed bees and the population in the next generation can be considered as the onlooker bees. To follow the concept of ABC, a new vector, x_{r_1} , which is called the base vector, chooses a food source with respect to

the probability that is computed from the fitness values of the current population. The probability value, p_i , of which x_i is chosen by a base vector x_{r_1} can be calculated by using the expression given in (14). After a base source x_{r_1} for a new vector is probabilistically chosen, both x_{r_2} and x_{r_3} are also chosen in the same manner as the terminal point and the other vector selections in the rank-DE. The target vector is created by a mutation formula of DE. The mutant vector u_i is created after the target vector is crossed with a randomly selected vector, and then the fitness value is computed. As in the ordinary DE, a greedy selection is applied between u_i and x_i . Hence, the new population contains better sources and positive feedback behavior appears. This idea can be expressed as pseudocode, as in Algorithm 2. Since the selection of x_{r_1} is the onlooker selection and the selections of x_{r_2} and x_{r_3} are brought from the rank-DE, then the algorithm is called onlooker and ranking-based vector selection.

The pseudocode of onlooker and ranking-based vector selection is shown in Algorithm 2. The differences between the original ranking-based and onlooker and ranking based selection are highlighted by “ \Leftarrow ”.

The function $\text{minprop}(\beta)$ is added to generalize the algorithm. Its output depends on the parameter β . The outcome can be either a constant value of $[0, 1)$ or a value of the uniform random function $\text{rndreal}[0, 1)$. The balance of the exploration and exploitation ability can be set by the parameter β . And the function is defined by

$$\begin{aligned} \text{minprop}(\beta) &= \begin{cases} \beta, & \text{if } \beta \text{ is a constant and } 0 \leq \beta < 1 \\ \text{rndreal}[0, 1), & \text{otherwise.} \end{cases} \end{aligned} \quad (15)$$

```

(1) Input: The target vector index  $i$ , the last index of onlooker  $r_1$ , and  $\beta$       ⇐
(2) Output: The selected vector indexes  $r_1, r_2, r_3$ 
(3)  $r_1 = r_1 + 1$ ; if  $r_1 > NP$  then  $r_1 = 1$ ; end if      ⇐
(4) while  $\text{minprop}(\beta) > p_{r_1}$  //onlooker-like selection      ⇐
(5)      $r_1 = r_1 + 1$ ; if  $r_1 > NP$  then  $r_1 = 1$ ; end if      ⇐
(6) end while
(7) Randomly select  $r_2 \in \{1, NP\}$  //terminal vector index
(8) while  $\text{rndreal}[0, 1] > p_{r_2}$  or  $r_2 == i$  or  $r_2 == r_1$  do
(9)     Randomly select  $r_2 \in \{1, NP\}$ 
(10) end while
(11) Randomly select  $r_3 \in \{1, NP\}$  //the other vector index
(12) while  $r_3 == r_2$  or  $r_3 == r_1$  or  $r_3 == i$  do
(13)     Randomly select  $r_3 \in \{1, NP\}$ 
(14) end while

```

ALGORITHM 2: Onlooker and ranking-based vector selection for DE.

```

(1) Randomly generate the initial population
(2) Evaluate the fitness for each individual in the population
(3) while the maximum generation  $G$  is not reached do
(4)     Sort and rank the fitness values of population according to (13)
(5)     Calculate the selection probability for each individual according to (14)
(6)      $r_1 = 0$       ⇐
(7)     for  $i = 1$  to  $NP$  do
(8)         Select  $r_1, r_2, r_3$  as shown in Algorithm 2 based on the current  $r_1$  and  $\beta$       ⇐
(9)          $j_{\text{rand}} = \text{rndint}[1, D]$ 
(10)        for  $j = 1$  to  $D$  do
(11)            if  $\text{rndreal}_j[0, 1] \leq CR$  or  $j$  is equal to  $j_{\text{rand}}$  then
(12)                 $u_{i,j} = x_{r_1,j} + F \cdot (x_{r_2,j} - x_{r_3,j})$ 
(13)            else
(14)                 $u_{i,j} = x_{i,j}$ 
(15)            end if
(16)        end for
(17)    end for
(18)    for  $i = 1$  to  $NP$  do
(19)        Evaluate the offspring  $u_i$ 
(20)        if  $f(u_i)$  is better than or equal to  $f(x_i)$  then
(21)            Replace  $x_i$  with  $u_i$ 
(22)        end if
(23)    end for
(24) end while

```

ALGORITHM 3: DE with onlooker and ranking-based mutation.

4.4. The DE with Onlooker-Ranking-Based Mutation Operator. The procedures in Sections 4.1, 4.2, and 4.3 are combined together to create a better DE algorithm. The parameter $0 \leq \beta < 1$ determines the fraction of the worse population to be eliminated. When $\beta = 0$ there is no worse population; each single vector in the current population will act as the base vector. If $0 < \beta < 1$, then each single vector having a probability less than β is a worse vector and will not be selected as the base vector. If β is not a constant or is outside $[0, 1)$, each single base vector is an onlooker bee. Accordingly, the name of the algorithm is Onlooker(β) Ranking-Base Differential Evolution ($O(\beta)R$ -DE). To achieve the global solution, a user can set a proper value for β to control

the balance of the exploration and exploitation abilities of the algorithm. The pseudocode of $O(\beta)R$ -DE is shown in Algorithm 3 and the differences between the rank-DE and $O(\beta)R$ -DE are highlighted by “ \Leftarrow ”.

5. Experiments and Results

5.1. Experimental Setup. The global multilevel thresholding problem deals with finding optimal thresholds within the range $[0, L - 1]$ that maximize the BCV function. The dimension of the optimization problem is the number of thresholds, n , and the search space is $[0, L-1]^n$. The parameter

β of $O(\beta)R$ -DE is $\text{rndreal}[0, 1)$ or is set to be one of 0.0, 0.1, ..., 0.9. The variation of the proposed $O(\beta)R$ -DE was implemented and compared with the existing metaheuristics that performed image thresholding, that is, PSO, DPSO, FODPSO, ABC, and several variations of DE algorithms. All the methods were programmed in Matlab R2013a and were run on a personal computer with a 3.4 GHz CPU, 8 GB RAM with Microsoft Windows 7 64-bit operating system. The experiments were conducted on 20 real images. The 19 images, namely, starfish, mountain, cactus, butterfly, circus, snow, palace, flower, wherry, waterfall, bird, police, ostrich, viaduct, fish, houses, mushroom, snow mountain, and snake, were taken from the Berkeley Segmentation Dataset and Benchmark [29]. The last image, namely, Riosanpablo, is a satellite image "New ISS Eyes see Rio San Pablo", March 1, 2013 (<http://visibleearth.nasa.gov/view.php?id=80561>). Each image has a unique gray level histogram. These original images and their histograms are depicted in Figure 1. An experiment of an image with a specific number of thresholds is called a "subproblem." The number of thresholds investigated in the experiments was 2, 3, ..., 16. Thus, there are 20×15 subproblems per algorithm. Each subproblem was repeated 50 times and each time is called a run.

To compare with PSO, ABC, and DEs algorithms, the objective function evaluation is computed for $NP \times N_i$, where NP is population size and N_i is the number of generations. A population of PSO and the DEs calls Otsu's function one time per generation. The population size in the PSO and DEs algorithms was set to 50. A bee in the ABC calls Otsu's function two times per generation; therefore their number of food sources were set to a half of the PSO's size, that is, 25. The stopping criteria were set by the maximum amount of generations G . In this experiment, G was set to 50, 100, 150, 200, 300, 400, 600, 800, 1000, 1500, 2000, 3000, 4000, 5000, and 6000 when n was 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16, respectively. For the PSO, DPSO, and FODPSO algorithms, the parameters were set as per the suggestion in [30] and is shown in Table 1. The other control parameter of the ABC algorithm, *limit*, was set to 50 [15]. The control parameters F and CR of the DE algorithms were set to 0.5 and 0.9, respectively [31, 32].

5.2. Comparison Strategies and Metrics. To compare the performance of different algorithms, there are three metrics: (1) the convergence rate of algorithms was compared by the average of generations (\overline{NG}), a lower \overline{NG} means a faster convergence rate; (2) the stability of algorithms was compared by the average of the success rate, (SR_{HM}), a higher SR_{HM} means higher stability; (3) the reliability was compared by the threshold value distortion measure (TVD), a lower TVD means higher reliability. The details of the three metrics are described as follows.

When all 50 runs of an algorithm performing on an image with a specific number of thresholds are terminated, the outcomes will be analyzed. Run r 'th is called a successful run if there is a generation of $t \leq G$ such that $BCV_r(t) \geq VTR$

TABLE 1: Essential parameters of the PSO, DPSO, and FODPSO taken from [30].

Parameter	PSO	DPSO	FODPSO
Population	50	50	50
ρ_1	1.5	1.5	1.5
ρ_2	1.5	1.5	1.5
W	1.2	1.2	1.2
V_{\max}	2	2	2
V_{\min}	-2	-2	-2
x_{\max}	255	255	255
x_{\min}	0	0	0
Min population	—	10	10
Max population	—	50	50
No. of swarms	—	4	4
Min swarms	—	2	2
Max swarms	—	6	6
Stagnancy	—	10	10
Fractional coefficient	—	—	0.75

and the number of generations (NG) of the successful run is recorded. Thus, the number can be defined by

$$NG_r = \underset{t}{\text{ArgMin}} (BCV_r(t) \geq VTR),$$

if r is a successful run and otherwise undefined. (16)

The average of NG_r from those successful runs is represented by \overline{NG} as follows:

$$\overline{NG} = \frac{1}{\text{number of successful runs}} \sum_{\text{All successful runs}} NG_r. \quad (17)$$

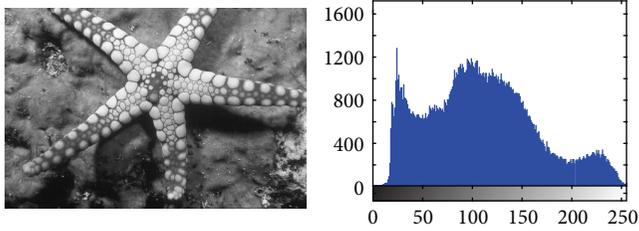
The ratio of success rate (SR) for which the algorithm succeeds to reach the VTR for each subproblem is computed as

$$SR = \frac{\text{number of successful runs}}{\text{total number of runs}}. \quad (18)$$

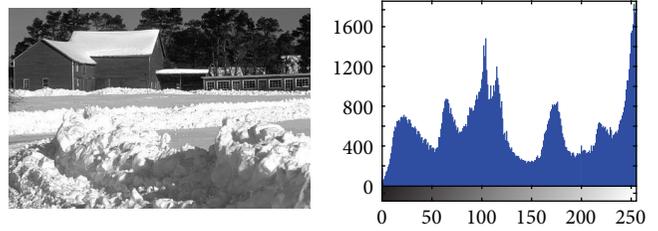
The experiments were conducted on 20 images. The arithmetic mean (AM) of \overline{NG} (\overline{NG}_{AM}) over the entire set of images with a specific number of thresholds is calculated as

$$\overline{NG}_{AM} = \frac{1}{N} \sum_{\text{All images}} \overline{NG}, \quad (19)$$

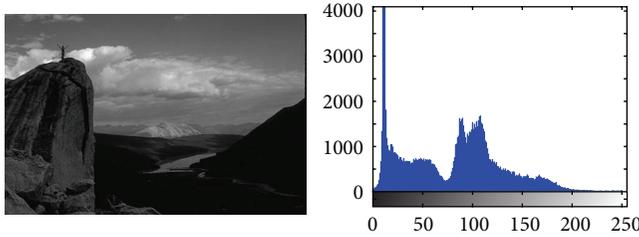
where N is the total number of images. \overline{NG}_{AM} is shown in Table 3. The worst-case scenario is that there is no successful run for a subproblem; this subproblem is called an "unsuccessful subproblem." If an algorithm encounters this scenario, the subproblem will be grouped by its number of thresholds and the number of images in the group will be counted and assigned to x . These scenarios will be represented by $NA(x)$, as shown in Tables 3 and 4.



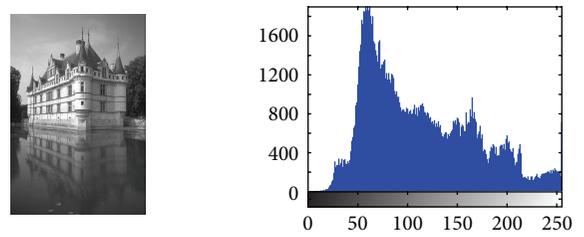
(1) Starfish (481 × 321)



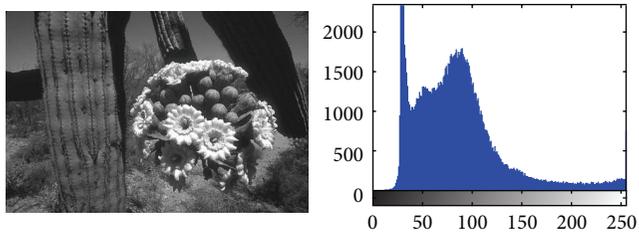
(6) Snow (481 × 321)



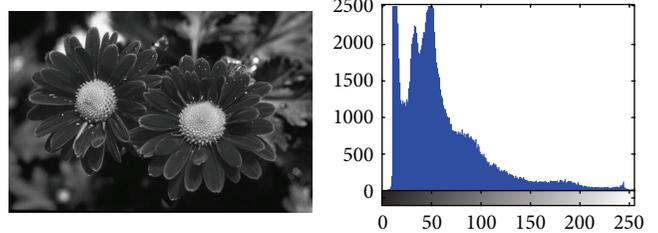
(2) Mountain (481 × 321)



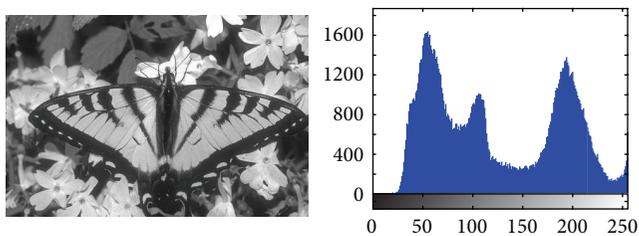
(7) Palace (321 × 481)



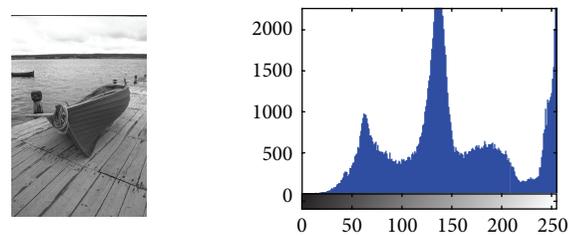
(3) Cactus (481 × 321)



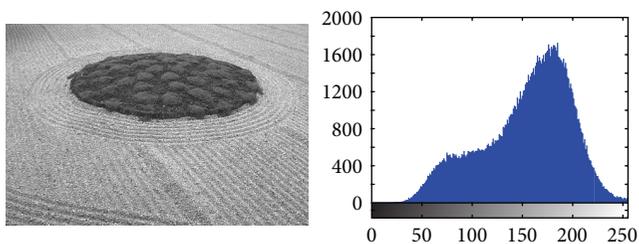
(8) Flower (481 × 321)



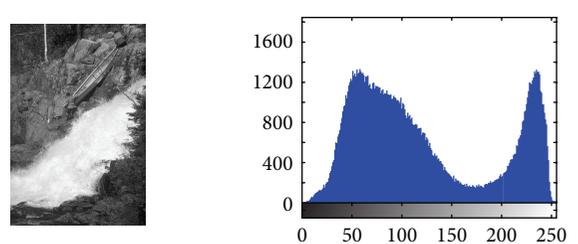
(4) Butterfly (481 × 321)



(9) Wherry (321 × 481)



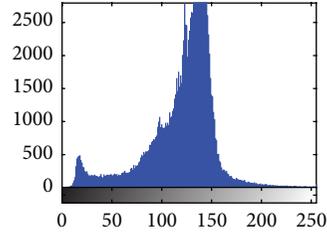
(5) Circus (481 × 321)



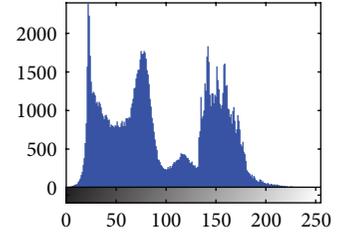
(10) Waterfall (321 × 481)

(a)

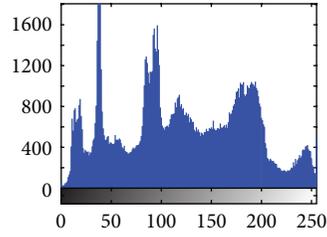
FIGURE 1: Continued.



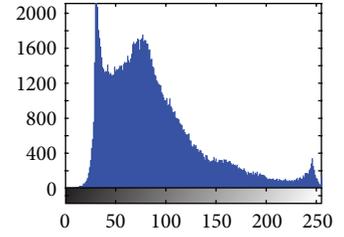
(11) Bird (321 × 481)



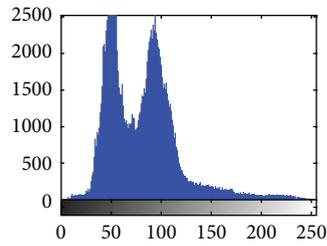
(16) Houses (481 × 321)



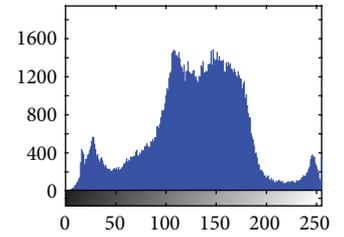
(12) Police (321 × 481)



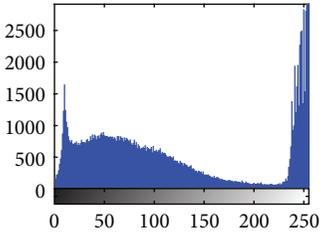
(17) Mushroom (321 × 481)



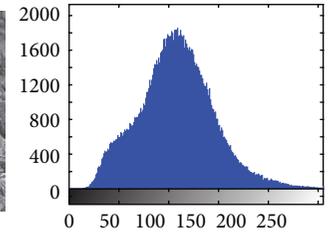
(13) Ostrich (321 × 481)



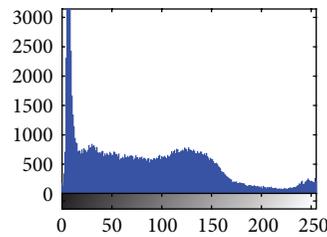
(18) Snow mountain (321 × 481)



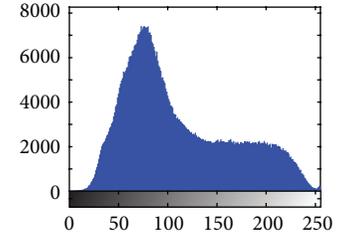
(14) Viaduct (481 × 321)



(19) Snake (481 × 321)



(15) Fish (481 × 321)



(20) Riosanpablo (720 × 944)

(b)

FIGURE 1: The test images and corresponding histograms.

The average of the success rate over the entire dataset with a specific number of thresholds (SR_{HM}) is averaged by the Harmonic mean, HM, as follows:

$$SR_{HM} = \frac{N}{\sum_{\text{All images}} (1/SR)}. \quad (20)$$

The SR_{HM} is very important in measuring the stability of an algorithm and it means the ratio of runs that are achieving the target solution. Because the evolutionary methods are based on stochastic searching algorithms, the solutions are not the same in each run of the algorithm and depend on the search ability of the algorithm. Therefore, the SR_{HM} is vital in evaluating the stability of the algorithms. The comparison of the stability gives us valuable information in terms of the ratio representing the success rates (SR_{HM}). A higher SR_{HM} means better stability of the algorithm.

An algorithm producing $SR_{HM} < 0.5$ means that more than 50 percent of the independent runs of the algorithm cannot reach the global solution. Thus, the algorithm that yields $SR_{HM} < 0.5$ should not be selected to solve the problem. The experiments were conducted for the number of thresholds varying from 2 to 16. These experiments contained the maximum number of thresholds such that the algorithm yields $SR_{HM} \geq 0.5$, which is represented by $n_{0.5}$ in Table 4. Furthermore, the experiments also contained the maximum number of thresholds that the algorithm can solve; and above this value there was the case such that all 50 runs of some subproblems missed the VTR. This number is represented by n_{max} . In this case the success rate was zero and the associated SR_{HM} was zero too. And the definitions of the two values are presented in (21)

$$\begin{aligned} n_{0.5} &= \max(\{n \mid n = \text{number of thresholds that} \\ &\quad \text{has } SR_{HM} \geq 0.5\}) \\ n_{max} &= \min(\{n \mid n = \text{number of thresholds that} \\ &\quad \text{has } SR_{HM} = 0\}) - 1. \end{aligned} \quad (21)$$

Let n be the number of thresholds. The reliability of a solution is measured by threshold value distortion measure (TVD) and is computed as

$$\begin{aligned} TVD &= \frac{\sum_{\text{All images}} \sum_{r=1}^{\text{run}} \sum_{i=1}^n |T_{ri}^* - T_{ri}^m|}{1 + \sum_{\text{All images}} \sum_{r=1}^{\text{run}} \sum_{i=1}^n 1_{\{T_{ri}^* \neq T_{ri}^m\}}} \\ &\quad \times (1 - SR) \times 100, \end{aligned} \quad (22)$$

where T^* is the threshold value producing the VTR, T^m is the threshold value obtained from the algorithm, and $1_{\{T_{ri}^* \neq T_{ri}^m\}}$ is the indicator function, which is equal to 1 when $T_{ri}^* \neq T_{ri}^m$ and is zero otherwise. TVD is zero if the algorithm can reach the VTR in every run. The lower the TVD the more reliable the algorithm is.

5.2.1. The Value to Reach (VTR). Following the completion of all of the experiments the best values of the between-class variance and the corresponding thresholds were collected

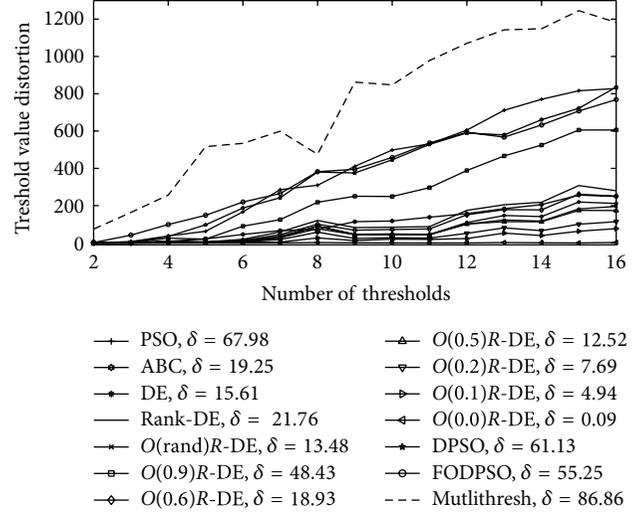


FIGURE 2: The threshold value distortion (TVD) of algorithms versus number of thresholds.

and are shown in Table 6. The results are shown image by image and the numbers of thresholds vary from 2 to 16. The between-class variance values in column 3 are used as the VTR values.

5.2.2. Results Produced by Local Search Method. The multi-thresh function of the Matlab toolbox was conducted on the same images and number of thresholds as the other search methods. The capabilities of solving the optimal solution between a local search and a global search will be discussed here. This is the reason we focused on the global search, that is, the proposed $O(0.0)R-DE$ algorithm. Table 2 shows the between-class variances and threshold values of the “mountain” image. These values were the best outcomes of 50 runs produced by the multithresh function in the Matlab R2013a toolbox and by the proposed $O(0.0)R-DE$ algorithm. The terminated condition of the multithresh function was set by “MaxFunEvals” = 500000. That is the multithresh function performs more function calls than that of the $O(0.0)R-DE$ algorithm. It can be seen from columns 3 and 5 that all the BCVs produced by the $O(0.0)R-DE$ algorithm are better than the BCVs produced by the multithresh function; the difference of the BCVs is shown in column 7. The differences in the thresholds from the two algorithms, shown in column 8, tended to be large if the number of thresholds increased.

Figure 2 shows the graph of the TVD of all the images and thresholds. These results are in the same pattern of the results of the “mountain” image in Table 2. That means the ability to search for the optimal solution of the proposed global search algorithm is higher than that of the multithresh function, especially when the number of thresholds is large. This goes to illustrate the difficulty of the problem. The problem with this kind is that it can be multimodal [33] or can be a nearly flat top surface [34]. The multithresh function solves the problem by performing the Nelder-Mead Simplex

TABLE 2: The best values of the between-class variance and thresholds of the ‘mountain’ image produced by the multithresh function of the Matlab toolbox and by the proposed $O(0.0)R$ -DE algorithm. The number of thresholds varies from 2 to 16.

Image (1)	n (2)	Between- class variance (3)	Produced by multithresh		Produced by the $O(0.0)R$ -DE algorithm		Objdiff (7) = (5) - (3)	Threshold (8) = $\sum (4)-(6) $
			Thresholds (4)	Between- class variance (5)	Thresholds (6)	Between- class variance (5)		
Mountain	2	2372.886	60, 127	2372.923	61, 128	0.037	2	
	3	2495.852	32, 76, 130	2496.113	33, 77, 131	0.261	3	
	4	2551.778	32, 72, 109, 145	2551.955	33, 73, 109, 147	0.177	4	
	5	2580.154	31, 68, 98, 124, 158	2580.336	32, 69, 99, 125, 159	0.182	5	
	6	2588.264	35, 75, 98, 118, 152, 175	2596.956	24, 46, 74, 101, 126, 160	8.692	122	
	7	2598.800	33, 70, 98, 118, 141, 166, 207	2608.807	24, 46, 73, 98, 119, 145, 175	10.007	153	
	8	2605.785	31, 68, 91, 105, 121, 140, 163, 192	2616.294	24, 46, 73, 97, 115, 135, 160, 191	10.509	70	
	9	2619.512	27, 52, 75, 95, 111, 129, 148, 168, 197	2622.314	20, 37, 54, 76, 98, 116, 136, 160, 191	2.802	114	
	10	2620.392	23, 46, 73, 97, 116, 135, 157, 175, 204, 232	2627.194	20, 36, 53, 74, 93, 106, 121, 140, 163, 194	6.802	258	
	11	2625.275	23, 45, 72, 93, 105, 119, 137, 156, 175, 204, 228	2630.496	20, 36, 53, 74, 93, 105, 118, 134, 152, 172, 201	5.221	199	
	12	2629.641	22, 39, 58, 78, 95, 109, 123, 139, 156, 177, 207, 247	2633.189	18, 31, 45, 59, 76, 93, 105, 118, 134, 152, 172, 201	3.548	246	
	13	2631.372	20, 38, 54, 74, 94, 107, 121, 139, 157, 173, 193, 224, 248	2635.290	18, 31, 45, 59, 76, 92, 103, 115, 129, 144, 161, 179, 207	3.918	283	
	14	2633.865	19, 36, 53, 74, 91, 103, 115, 129, 144, 160, 174, 190, 217, 245	2637.088	17, 29, 42, 56, 72, 86, 96, 106, 117, 130, 145, 161, 179, 207	3.223	307	
	15	2635.190	19, 33, 50, 69, 85, 96, 107, 121, 137, 152, 167, 178, 190, 212, 240	2638.449	17, 28, 39, 50, 61, 75, 88, 97, 107, 118, 131, 146, 162, 179, 207	3.259	351	
	16	2636.357	16, 30, 45, 59, 77, 93, 105, 118, 133, 148, 162, 176, 192, 209, 230, 244	2639.616	17, 27, 38, 49, 60, 74, 86, 95, 104, 113, 123, 135, 149, 164, 181, 209	3.259	415	

TABLE 3: The average of mean number of generation (\overline{NG}_{AM}) and ranks of the methods.

No.	FODPSO	DPSO	PSO	ABC	Rank-DE	O(rand)R-DE	O(0.9)R-DE	O(0.8)R-DE	O(0.7)R-DE	O(0.6)R-DE	O(0.5)R-DE	O(0.4)R-DE	O(0.3)R-DE	O(0.2)R-DE	O(0.1)R-DE	O(0.0)R-DE
$n = 2$	18.337	15.713	13.218	13.202	10.538	14.713	10.117	6.394	6.903	7.636	8.208	9.147	9.901	10.709	11.631	20.850
$G = 50$	Rank 16	Rank 15	Rank 13	Rank 12	Rank 9	Rank 14	Rank 8	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6	Rank 7	Rank 10	Rank 17
$n = 3$	35.200	36.210	27.479	33.210	19.488	29.049	18.958	10.492	12.296	13.587	14.807	16.969	17.107	18.100	19.546	44.192
$G = 100$	Rank 15	Rank 16	Rank 12	Rank 14	Rank 9	Rank 13	Rank 8	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6	Rank 7	Rank 10	Rank 17
$n = 4$	45.391	58.683	49.596	63.251	31.060	48.130	29.595	14.733	18.526	20.885	22.815	26.632	26.456	28.058	30.175	72.100
$G = 150$	Rank 12	Rank 15	Rank 14	Rank 16	Rank 10	Rank 13	Rank 8	Rank 1	Rank 2	Rank 3	Rank 4	Rank 6	Rank 5	Rank 7	Rank 9	Rank 17
$n = 5$	55.664	81.877	87.441	108.131	44.157	69.088	41.283	19.508	24.524	28.097	30.963	36.935	36.748	38.647	41.455	100.693
$G = 200$	Rank 12	Rank 14	Rank 15	Rank 17	Rank 10	Rank 13	Rank 8	Rank 1	Rank 2	Rank 3	Rank 4	Rank 6	Rank 5	Rank 7	Rank 9	Rank 16
$n = 6$	NA(2)	100.664	141.937	157.328	61.935	99.700	56.405	24.623	32.687	38.167	42.608	51.003	50.281	52.089	55.729	138.716
$G = 300$	Rank 17	Rank 13	Rank 15	Rank 16	Rank 11	Rank 12	Rank 9	Rank 1	Rank 2	Rank 3	Rank 4	Rank 6	Rank 5	Rank 7	Rank 8	Rank 14
$n = 7$	NA(1)	NA(1)	192.267	236.928	86.938	142.096	75.340	29.763	41.691	49.408	56.193	67.619	67.063	69.696	74.389	184.371
$G = 400$	Rank 17	Rank 16	Rank 14	Rank 15	Rank 11	Rank 12	Rank 9	Rank 1	Rank 2	Rank 3	Rank 4	Rank 6	Rank 5	Rank 7	Rank 8	Rank 13
$n = 8$	NA(7)	NA(6)	NA(2)	NA(1)	117.092	198.986	100.944	35.967	51.573	62.263	74.523	91.836	89.666	95.198	102.552	244.880
$G = 600$	Rank 17	Rank 16	Rank 15	Rank 14	Rank 11	Rank 12	Rank 8	Rank 1	Rank 2	Rank 3	Rank 4	Rank 6	Rank 5	Rank 7	Rank 9	Rank 13
$n = 9$	NA(11)	NA(13)	NA(12)	NA(3)	151.004	264.253	125.299	41.228	63.742	79.432	90.201	113.530	110.416	122.555	137.538	299.492
$G = 800$	Rank 17	Rank 16	Rank 15	Rank 14	Rank 11	Rank 12	Rank 9	Rank 1	Rank 2	Rank 3	Rank 4	Rank 6	Rank 5	Rank 7	Rank 8	Rank 13
$n = 10$	NA(11)	NA(16)	NA(16)	NA(7)	NA(1)	321.373	147.770	NA(3)	73.564	91.121	105.020	132.968	128.999	132.691	141.825	356.546
$G = 1000$	Rank 17	Rank 16	Rank 15	Rank 14	Rank 12	Rank 10	Rank 8	Rank 13	Rank 1	Rank 2	Rank 3	Rank 6	Rank 4	Rank 5	Rank 7	Rank 11
$n = 11$	NA(16)	NA(17)	NA(19)	NA(9)	NA(1)	405.090	185.020	NA(3)	NA(2)	NA(2)	127.135	162.822	157.184	160.123	175.152	434.877
$G = 1500$	Rank 17	Rank 16	Rank 15	Rank 14	Rank 12	Rank 8	Rank 6	Rank 13	Rank 10	Rank 10	Rank 1	Rank 4	Rank 2	Rank 3	Rank 5	Rank 9
$n = 12$	NA(18)	NA(20)	NA(20)	NA(9)	278.743	535.542	229.965	NA(4)	NA(5)	135.266	160.047	205.266	197.386	203.814	222.094	574.104
$G = 2000$	Rank 17	Rank 16	Rank 15	Rank 14	Rank 12	Rank 8	Rank 6	Rank 13	Rank 11	Rank 10	Rank 1	Rank 4	Rank 2	Rank 3	Rank 5	Rank 9
$n = 13$	NA(20)	NA(19)	NA(20)	NA(14)	NA(3)	NA(3)	275.860	NA(12)	NA(5)	NA(1)	NA(1)	NA(1)	NA(1)	261.522	316.468	726.521
$G = 3000$	Rank 17	Rank 16	Rank 15	Rank 14	Rank 12	Rank 9	Rank 2	Rank 13	Rank 11	Rank 10	Rank 6	Rank 6	Rank 1	Rank 3	Rank 4	Rank 5
$n = 14$	NA(20)	NA(20)	NA(20)	NA(14)	NA(4)	NA(4)	330.194	NA(11)	NA(8)	NA(4)	NA(1)	NA(2)	296.414	331.626	385.951	889.711
$G = 4000$	Rank 17	Rank 16	Rank 15	Rank 14	Rank 12	Rank 9	Rank 1	Rank 13	Rank 11	Rank 10	Rank 7	Rank 8	Rank 6	Rank 5	Rank 3	Rank 4
$n = 15$	NA(20)	NA(20)	NA(20)	NA(19)	NA(13)	NA(5)	399.228	NA(11)	NA(10)	NA(8)	NA(6)	363.506	362.665	383.788	417.958	1086.001
$G = 5000$	Rank 17	Rank 16	Rank 15	Rank 14	Rank 13	Rank 8	Rank 1	Rank 12	Rank 11	Rank 10	Rank 9	Rank 7	Rank 6	Rank 5	Rank 2	Rank 4
$n = 16$	NA(20)	NA(20)	NA(20)	NA(18)	NA(10)	NA(8)	NA(4)	NA(18)	NA(11)	NA(9)	NA(5)	NA(1)	NA(2)	447.570	552.856	1374.438
$G = 6000$	Rank 17	Rank 16	Rank 15	Rank 14	Rank 12	Rank 9	Rank 4	Rank 13	Rank 11	Rank 10	Rank 8	Rank 6	Rank 7	Rank 5	Rank 1	Rank 3
r_{max}^*	5,	6,	7,	7,	9,	12,	15,	9,	10,	10,	12,	12,	12,	13,	16,	16,
Avg for NA(x)	NA(146),	NA(152),	NA(149),	NA(94),	NA(32),	NA(20),	NA(4),	NA(62),	NA(41),	NA(24),	NA(13),	NA(4),	NA(3),	NA(1),	NA(0),	NA(0),
$n = 2$ to 16	\overline{NG}_{AM}	58.629	85.323	102.008	88.995	193.456	144.713	22.589	36.167	52.586	66.593	106.519	119.204	143.971	167.708	190.391
Rank	Rank 17	Rank 16	Rank 15	Rank 14	Rank 12	Rank 9	Rank 4	Rank 13	Rank 11	Rank 10	Rank 8	Rank 7	Rank 6	Rank 5	Rank 1	Rank 2

TABLE 4: The average success rate (SR_{HIM}) and ranks of the methods.

No	Multi-thresh	FODPSO	DPSO	PSO	ABC	Rank-DE	O(rand)R-DE	O(0.9)R-DE	O(0.8)R-DE	O(0.7)R-DE	O(0.6)R-DE	O(0.5)R-DE	O(0.4)R-DE	O(0.3)R-DE	O(0.2)R-DE	O(0.1)R-DE	O(0.0)R-DE	
2	SR _{HIM} Rank	0 17	0.977 1	1 1	1 1	1 1	0.999 1	0.999 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	
3	SR _{HIM} Rank	0.000 17	0.635 16	0.972 16	0.991 13	0.979 15	0.998 8	1.000 1	0.993 12	0.999 6	0.997 10	0.998 8	0.999 6	1.000 1	1.000 1	1.000 1	1.000 1	
4	SR _{HIM} Rank	0.000 17	0.379 17	0.742 16	0.934 13	0.833 15	0.987 7	0.995 3	0.986 12	0.972 10	0.971 11	0.986 8	0.993 4	0.992 5	0.991 6	0.998 1	0.997 2	
5	SR _{HIM} Rank	0.000 17	0.185 17	0.391 16	0.767 14	0.706 15	0.936 10	0.977 3	0.878 11	0.874 12	0.944 9	0.950 8	0.959 7	0.968 4	0.960 5	0.985 2	1.000 1	
6	SR _{HIM} Rank	0.000 17	0.000 17	0.187 16	0.433 15	0.453 14	0.863 10	0.968 3	0.601 13	0.826 11	0.878 9	0.934 6	0.922 7	0.944 4	0.935 5	0.970 2	0.999 1	
7	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.153 15	0.164 14	0.696 10	0.884 4	0.835 8	0.675 11	0.741 9	0.840 7	0.851 6	0.883 5	0.914 3	0.949 2	0.998 1	
8	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.000 14	0.000 14	0.300 11	0.458 9	0.530 7	0.351 10	0.502 8	0.592 6	0.628 5	0.724 4	0.757 3	0.838 2	0.972 1	
9	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.000 14	0.000 10	0.495 10	0.695 7	0.681 8	0.372 11	0.548 9	0.721 6	0.725 5	0.769 4	0.815 3	0.852 2	0.999 1	
10	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.000 14	0.000 12	0.000 12	0.381 6	0.375 7	0.218 10	0.311 9	0.409 5	0.370 8	0.460 4	0.578 3	0.686 2	0.998 1	
11	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.000 14	0.000 12	0.000 12	0.271 7	0.267 8	0.000 10	0.144 9	0.407 5	0.316 6	0.424 4	0.584 3	0.704 2	0.999 1	
12	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.000 14	0.000 12	0.090 8	0.270 8	0.423 5	0.089 10	0.177 9	0.400 7	0.407 6	0.501 4	0.641 3	0.810 2	1.000 1	
13	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.000 14	0.000 12	0.000 6	0.000 6	0.151 5	0.000 10	0.000 6	0.000 6	0.000 6	0.254 4	0.267 3	0.381 2	0.974 1	
14	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.000 14	0.000 12	0.000 7	0.000 4	0.102 4	0.000 10	0.000 7	0.000 7	0.110 6	0.000 5	0.253 3	0.494 2	0.993 1	
15	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.000 14	0.000 12	0.000 8	0.000 4	0.090 4	0.000 10	0.000 8	0.095 7	0.117 6	0.149 5	0.288 3	0.466 2	1.000 1	
16	SR _{HIM} Rank	0.000 17	0.000 17	0.000 16	0.000 14	0.000 12	0.000 6	0.000 4	0.000 4	0.000 10	0.000 6	0.000 6	0.000 6	0.110 5	0.192 3	0.288 2	0.977 1	
Avg, for n_0, s	<2,	3,	4,	5,	5,	7,	9,	9,	9,	7,	9,	9,	9,	9,	12,	12,	16,	16,
Avg, for n_{max}	<2,	5,	6,	7,	7,	9,	12,	15,	15,	10,	12,	12,	12,	13,	16,	16,	16,	16,
$n = 2$ to SR _{HIM}	0.000	0.376	0.443	0.454	0.464	0.665	0.554	0.304	0.263	0.313	0.530	0.420	0.654	0.619	0.498	0.656	0.994	
16	Rank	18	17	16	15	14	12	8	4	10	10	9	7	5	3	2	1	

TABLE 5: The average of threshold value distortion (TVD) and ranks of the methods.

No	Multi-thresh	FODPSO	DPSO	PSO	ABC	Rank-DE	DE	O(rand)R-DE	O(0.9)R-DE	O(0.8)R-DE	O(0.7)R-DE	O(0.6)R-DE	O(0.5)R-DE	O(0.4)R-DE	O(0.3)R-DE	O(0.2)R-DE	O(0.1)R-DE	O(0.0)R-DE
2	TVD Rank	74.153 18	2.217 17	0.000 1	0.000 1	0.000 1	0.000 1	0.050 16	0.000 1	0.000 1	0.000 1	0.000 1	0.000 1	0.000 1	0.000 1	0.000 1	0.000 1	0.000 1
3	TVD Rank	163.333 18	41.753 17	7.424 16	1.510 14	0.150 8	0.000 1	1.004 13	0.004 13	0.519 12	0.075 6	0.246 10	0.171 9	0.075 6	0.000 1	0.000 1	0.000 1	0.000 1
4	TVD Rank	257.083 18	98.848 17	34.770 16	8.716 12	7.340 9	1.428 4	2.184 6	25.499 14	13.609 13	7.318 8	7.360 10	7.398 11	4.303 7	1.672 5	0.728 3	0.130 1	0.257 2
5	TVD Rank	516.582 18	148.344 17	97.216 16	62.319 15	5.373 10	1.864 3	3.241 5	20.441 13	10.064 11	10.422 12	4.584 10	3.904 8	3.289 6	2.441 4	3.385 7	1.195 2	0.000 1
6	TVD Rank	534.226 18	219.437 17	187.620 16	166.775 15	45.728 13	2.528 3	7.761 8	89.938 14	43.666 12	30.923 11	13.010 9	6.524 7	6.023 5	4.136 4	6.129 6	2.421 2	0.050 1
7	TVD Rank	599.250 18	265.029 16	242.194 15	285.862 17	66.259 12	27.976 5	35.481 8	125.040 14	75.011 13	49.851 10	44.610 9	28.039 6	29.841 7	26.075 4	17.950 3	4.776 2	0.150 1
8	TVD Rank	475.458 18	382.076 17	380.124 16	309.635 15	72.763 5	90.287 9	89.406 8	218.266 14	143.225 13	130.898 12	99.991 10	76.353 6	82.886 7	57.251 3	57.665 4	26.822 2	2.103 1
9	TVD Rank	862.458 18	394.744 16	375.229 15	410.710 17	113.892 12	44.407 7	46.539 8	250.470 14	154.662 13	96.417 11	68.197 9	44.223 6	35.333 5	26.627 4	23.406 3	12.757 2	0.780 1
10	TVD Rank	847.703 18	458.536 16	445.765 15	498.328 17	117.664 12	43.991 6	48.474 8	249.147 14	136.935 13	93.935 11	72.288 9	43.883 5	44.110 7	36.700 4	27.978 3	20.213 2	0.164 1
11	TVD Rank	977.976 18	536.258 17	528.329 16	531.211 15	137.941 12	44.944 6	44.973 7	296.513 14	152.195 13	104.092 11	75.115 9	44.461 5	47.564 8	34.944 4	26.231 3	18.782 2	0.083 1
12	TVD Rank	1070.300 18	592.750 16	589.931 15	605.280 17	157.169 12	109.138 8	101.282 7	388.491 14	258.312 13	196.098 12	150.932 9	101.112 6	82.365 5	79.966 4	51.281 3	23.429 2	0.000 1
13	TVD Rank	1142.808 18	567.442 15	579.495 16	711.382 17	183.451 10	147.251 8	123.438 7	466.508 14	299.040 13	216.656 12	178.394 9	114.244 5	114.491 6	96.104 4	81.076 3	52.578 2	1.975 1
14	TVD Rank	1148.641 18	632.482 15	660.908 16	770.319 17	205.714 10	141.431 8	116.716 7	524.783 14	325.936 13	241.997 12	177.121 9	113.426 6	104.073 5	88.593 4	65.908 3	39.647 2	0.564 1
15	TVD Rank	1245.529 18	707.835 15	722.126 16	816.158 17	256.477 9	219.556 8	182.464 7	605.623 14	428.370 13	343.439 12	261.522 10	174.700 6	166.309 5	138.710 4	100.296 3	62.940 2	0.000 1
16	TVD Rank	1182.989 18	768.714 15	835.747 16	828.004 17	249.042 9	211.614 8	195.676 7	605.790 14	416.711 13	311.314 12	251.408 10	174.118 6	167.316 5	133.922 4	113.776 3	76.661 2	2.539 1
Avg. for $n = 2$ to 16	Growth rate (δ) Rank	86.86 18	55.25 15	61.13 16	67.98 17	19.25 10	21.76 8	13.48 7	48.43 14	32.25 13	23.91 12	18.93 9	12.52 6	11.86 5	9.99 4	7.69 3	4.94 2	0.09 1

TABLE 6: The between-class variance criterion and the best thresholds for test images.

Image	n	Between-class variance (VTR)	Thresholds
Starfish	2	2546.885	85, 157
	3	2779.925	68, 119, 177
	4	2865.707	60, 101, 138, 187
	5	2912.859	52, 86, 117, 150, 194
	6	2941.728	47, 77, 105, 132, 162, 201
	7	2960.158	44, 71, 95, 118, 142, 170, 206
	8	2972.356	43, 68, 90, 110, 131, 153, 180, 212
	9	2981.138	38, 58, 78, 97, 116, 136, 157, 183, 214
	10	2988.206	37, 56, 75, 93, 110, 128, 146, 167, 192, 219
	11	2993.348	35, 53, 71, 88, 103, 119, 135, 152, 172, 196, 221
	12	2997.352	34, 51, 68, 84, 98, 112, 127, 142, 158, 177, 200, 223
	13	3000.480	33, 48, 64, 79, 93, 106, 119, 133, 147, 162, 181, 203, 225
	14	3003.076	32, 47, 62, 76, 89, 101, 114, 127, 140, 154, 169, 187, 207, 227
	15	3005.235	30, 43, 56, 70, 83, 95, 107, 119, 131, 143, 156, 171, 189, 209, 228
	16	3007.060	30, 42, 55, 68, 80, 92, 103, 114, 126, 138, 150, 163, 178, 195, 213, 230
	Mountain	2	2372.923
3		2496.113	33, 77, 131
4		2551.955	33, 73, 109, 147
5		2580.336	32, 69, 99, 125, 159
6		2596.956	24, 46, 74, 101, 126, 160
7		2608.807	24, 46, 73, 98, 119, 145, 175
8		2616.294	24, 46, 73, 97, 115, 135, 160, 191
9		2622.314	20, 37, 54, 76, 98, 116, 136, 160, 191
10		2627.194	20, 36, 53, 74, 93, 106, 121, 140, 163, 194
11		2630.496	20, 36, 53, 74, 93, 105, 118, 134, 152, 172, 201
12		2633.189	18, 31, 45, 59, 76, 93, 105, 118, 134, 152, 172, 201
13		2635.290	18, 31, 45, 59, 76, 92, 103, 115, 129, 144, 161, 179, 207
14		2637.088	17, 29, 42, 56, 72, 86, 96, 106, 117, 130, 145, 161, 179, 207
15		2638.449	17, 28, 39, 50, 61, 75, 88, 97, 107, 118, 131, 146, 162, 179, 207
16		2639.616	17, 27, 38, 49, 60, 74, 86, 95, 104, 113, 123, 135, 149, 164, 181, 209
Cactus		2	1816.448
	3	1970.112	64, 106, 173
	4	2042.275	55, 87, 125, 187
	5	2080.884	49, 76, 102, 138, 196
	6	2104.147	46, 70, 93, 119, 157, 208
	7	2119.670	44, 65, 85, 105, 131, 168, 215
	8	2129.358	42, 60, 77, 94, 113, 138, 174, 218
	9	2136.162	41, 58, 74, 89, 105, 125, 152, 186, 224
	10	2141.579	40, 56, 71, 85, 99, 115, 135, 161, 193, 228
	11	2145.397	39, 53, 66, 79, 91, 104, 119, 139, 165, 196, 229
	12	2148.282	38, 51, 64, 76, 88, 100, 114, 131, 152, 176, 204, 233
	13	2150.740	37, 49, 61, 72, 83, 94, 105, 118, 135, 155, 178, 205, 233
	14	2152.626	36, 47, 58, 68, 78, 88, 98, 109, 122, 138, 158, 181, 207, 234
	15	2154.162	36, 46, 56, 66, 76, 85, 94, 104, 115, 128, 144, 164, 187, 212, 237
	16	2155.471	36, 46, 56, 66, 75, 84, 93, 102, 112, 124, 138, 155, 174, 195, 217, 239

TABLE 6: Continued.

Image	n	Between-class variance (VTR)	Thresholds
Butterfly	2	3873.222	84, 155
	3	3990.855	81, 144, 199
	4	4051.357	77, 121, 167, 207
	5	4092.929	60, 89, 129, 172, 209
	6	4119.448	59, 87, 122, 161, 193, 221
	7	4135.937	53, 74, 98, 128, 165, 195, 222
	8	4148.556	53, 73, 96, 123, 156, 183, 203, 228
	9	4156.528	52, 72, 94, 118, 144, 170, 190, 207, 231
	10	4163.794	48, 63, 80, 99, 121, 147, 172, 191, 208, 231
	11	4168.489	47, 62, 79, 98, 118, 141, 164, 183, 198, 213, 234
	12	4172.517	46, 59, 73, 89, 104, 122, 144, 167, 185, 199, 214, 235
	13	4175.487	46, 59, 72, 87, 102, 118, 137, 157, 175, 189, 202, 216, 236
	14	4177.893	44, 55, 66, 79, 93, 106, 121, 141, 161, 178, 191, 203, 217, 237
	15	4179.908	44, 55, 66, 78, 92, 105, 119, 137, 156, 173, 186, 197, 208, 221, 239
	16	4181.559	42, 52, 61, 71, 83, 95, 107, 121, 139, 157, 173, 186, 197, 208, 221, 239
	Circus	2	1651.257
3		1760.512	105, 150, 187
4		1817.487	93, 132, 165, 195
5		1850.243	87, 122, 152, 177, 203
6		1870.083	82, 113, 141, 164, 185, 208
7		1883.966	77, 104, 129, 151, 171, 190, 212
8		1893.450	73, 98, 122, 143, 161, 178, 195, 216
9		1900.592	70, 92, 114, 134, 152, 168, 183, 199, 219
10		1905.992	68, 89, 109, 128, 145, 160, 174, 188, 203, 222
11		1909.887	66, 86, 105, 123, 139, 153, 166, 179, 192, 206, 225
12		1913.079	63, 81, 98, 114, 130, 145, 158, 170, 182, 194, 208, 226
13		1915.676	62, 79, 95, 111, 126, 140, 152, 164, 175, 186, 197, 210, 228
14		1917.756	61, 78, 94, 109, 123, 136, 148, 159, 170, 180, 190, 201, 214, 231
15		1919.524	59, 74, 88, 102, 115, 128, 140, 151, 161, 171, 181, 191, 202, 214, 231
16		1920.958	58, 73, 87, 100, 113, 126, 138, 149, 159, 169, 178, 187, 196, 206, 218, 234
Snow		2	5261.705
	3	5624.289	71, 139, 207
	4	5729.116	50, 92, 144, 208
	5	5785.138	49, 91, 140, 192, 231
	6	5819.333	45, 81, 111, 148, 194, 232
	7	5835.770	43, 76, 101, 127, 159, 196, 232
	8	5850.190	30, 55, 83, 107, 133, 163, 197, 233
	9	5862.437	30, 55, 82, 106, 129, 157, 185, 211, 237
	10	5870.284	29, 52, 75, 94, 111, 132, 159, 186, 212, 237
	11	5875.817	29, 52, 75, 94, 111, 132, 158, 183, 206, 227, 244
	12	5880.335	29, 52, 74, 93, 109, 127, 149, 169, 188, 209, 228, 244
	13	5884.513	23, 39, 57, 76, 94, 110, 128, 150, 170, 188, 209, 228, 244
	14	5887.355	22, 37, 54, 70, 84, 98, 112, 129, 151, 170, 188, 209, 228, 244
	15	5889.953	21, 36, 53, 69, 83, 97, 110, 124, 141, 159, 175, 192, 211, 229, 245
	16	5891.998	21, 36, 53, 69, 83, 97, 110, 124, 141, 159, 174, 189, 207, 222, 236, 248

TABLE 6: Continued.

Image	n	Between-class variance (VTR)	Thresholds
Palace	2	2623.440	99, 165
	3	2791.488	84, 132, 186
	4	2860.98	70, 103, 143, 191
	5	2908.165	69, 101, 138, 177, 218
	6	2934.330	64, 89, 117, 147, 181, 220
	7	2953.745	54, 75, 99, 126, 153, 183, 220
	8	2966.250	50, 69, 89, 111, 134, 158, 185, 221
	9	2974.719	47, 65, 81, 100, 120, 141, 163, 188, 222
	10	2981.875	47, 64, 80, 98, 117, 137, 157, 178, 199, 226
	11	2986.898	45, 61, 75, 90, 106, 123, 141, 159, 179, 200, 227
	12	2990.579	43, 58, 69, 82, 96, 111, 126, 143, 160, 180, 201, 227
	13	2993.809	43, 58, 69, 81, 95, 110, 125, 141, 157, 173, 190, 208, 231
	14	2996.112	43, 57, 68, 80, 94, 108, 123, 138, 153, 167, 182, 199, 218, 239
	15	2998.213	42, 56, 66, 77, 88, 100, 113, 126, 140, 154, 167, 182, 199, 218, 239
	16	2999.918	42, 55, 65, 75, 86, 98, 110, 123, 136, 149, 162, 176, 191, 205, 222, 241
	Flower	2	1489.281
3		1627.897	39, 77, 141
4		1685.956	36, 67, 105, 160
5		1715.220	28, 49, 75, 111, 164
6		1736.423	27, 47, 71, 102, 143, 192
7		1752.512	26, 43, 61, 83, 111, 151, 199
8		1761.968	25, 42, 58, 77, 99, 126, 161, 204
9		1768.354	24, 40, 54, 70, 88, 109, 135, 167, 207
10		1772.903	23, 37, 48, 60, 75, 92, 112, 138, 169, 208
11		1776.470	23, 36, 47, 58, 72, 87, 104, 124, 149, 177, 211
12		1779.106	22, 34, 44, 54, 65, 78, 92, 108, 128, 152, 179, 212
13		1781.251	20, 30, 39, 48, 58, 70, 83, 96, 112, 131, 154, 180, 213
14		1783.049	19, 29, 38, 47, 56, 66, 78, 90, 104, 120, 139, 161, 185, 215
15		1784.478	19, 29, 38, 46, 54, 63, 74, 85, 97, 111, 128, 147, 168, 190, 218
16		1785.641	19, 28, 37, 45, 53, 62, 72, 83, 94, 106, 120, 136, 155, 175, 196, 222
Wherry		2	3313.161
	3	3543.272	102, 161, 218
	4	3599.924	83, 121, 163, 218
	5	3634.708	81, 118, 152, 184, 224
	6	3656.048	72, 103, 130, 156, 186, 225
	7	3668.313	68, 94, 120, 139, 161, 189, 226
	8	3678.216	60, 83, 110, 132, 152, 175, 198, 230
	9	3686.001	56, 77, 100, 122, 138, 156, 179, 201, 231
	10	3691.730	56, 76, 98, 120, 136, 153, 174, 194, 219, 243
	11	3696.416	55, 74, 94, 114, 130, 142, 158, 178, 197, 222, 245
	12	3699.866	54, 72, 91, 110, 126, 138, 151, 168, 185, 202, 225, 246
	13	3702.619	52, 68, 83, 100, 117, 130, 140, 153, 169, 185, 202, 225, 246
	14	3705.033	52, 68, 83, 100, 117, 130, 140, 152, 167, 182, 197, 215, 235, 249
	15	3706.937	51, 66, 79, 94, 110, 123, 133, 142, 154, 169, 184, 199, 216, 235, 249
	16	3708.484	49, 63, 75, 89, 104, 118, 129, 138, 147, 159, 173, 186, 200, 217, 235, 249

TABLE 6: Continued.

Image	n	Between-class variance (VTR)	Thresholds
Waterfall	2	4512.801	88, 170
	3	4646.137	72, 115, 182
	4	4711.019	67, 103, 150, 204
	5	4752.267	58, 87, 119, 164, 212
	6	4777.063	53, 78, 104, 134, 176, 217
	7	4793.510	48, 70, 92, 116, 147, 186, 221
	8	4805.756	46, 66, 86, 107, 131, 163, 199, 226
	9	4814.097	43, 61, 79, 97, 117, 141, 173, 205, 228
	10	4820.724	40, 57, 73, 90, 108, 128, 152, 182, 210, 230
	11	4825.739	38, 54, 69, 84, 100, 117, 137, 162, 191, 215, 232
	12	4829.689	37, 53, 67, 81, 96, 111, 128, 148, 173, 198, 218, 233
	13	4832.826	35, 50, 63, 76, 89, 103, 117, 133, 153, 177, 201, 220, 234
	14	4835.371	32, 46, 58, 70, 82, 95, 108, 122, 138, 158, 182, 204, 221, 234
	15	4837.537	32, 46, 57, 68, 80, 92, 104, 117, 131, 148, 169, 191, 210, 224, 236
	16	4839.226	30, 43, 54, 64, 74, 85, 96, 108, 120, 134, 151, 172, 193, 211, 225, 236
	Bird	2	901.450
3		975.230	64, 111, 140
4		1027.509	61, 104, 131, 164
5		1051.482	54, 93, 119, 138, 169
6		1067.992	47, 82, 108, 127, 142, 172
7		1077.304	40, 70, 94, 113, 129, 143, 173
8		1086.005	39, 69, 93, 112, 128, 141, 159, 192
9		1091.341	37, 65, 88, 105, 119, 131, 142, 160, 193
10		1095.355	37, 64, 86, 103, 117, 129, 139, 149, 167, 200
11		1098.475	33, 56, 77, 93, 107, 119, 130, 140, 150, 169, 202
12		1100.749	33, 56, 77, 93, 107, 119, 129, 138, 146, 158, 178, 209
13		1102.639	31, 52, 71, 87, 100, 111, 121, 130, 139, 147, 159, 179, 210
14		1104.132	29, 48, 66, 82, 95, 107, 118, 127, 134, 141, 149, 161, 181, 211
15		1105.446	28, 45, 62, 77, 90, 101, 111, 120, 128, 135, 142, 150, 162, 182, 212
16		1106.500	28, 45, 62, 77, 90, 101, 111, 120, 128, 135, 141, 148, 157, 171, 191, 219
Police		2	3647.353
	3	3844.314	70, 135, 192
	4	3966.225	63, 112, 158, 209
	5	4013.875	61, 104, 140, 174, 214
	6	4047.198	32, 67, 106, 141, 175, 214
	7	4067.996	32, 67, 104, 133, 161, 186, 219
	8	4084.933	29, 52, 78, 106, 134, 162, 187, 219
	9	4094.705	29, 52, 78, 103, 125, 147, 169, 190, 220
	10	4101.702	29, 52, 78, 102, 123, 143, 165, 184, 203, 228
	11	4108.018	28, 49, 71, 90, 107, 126, 146, 166, 185, 204, 229
	12	4112.304	28, 49, 71, 89, 105, 122, 139, 157, 174, 189, 207, 231
	13	4115.165	28, 46, 62, 78, 92, 106, 123, 140, 158, 174, 189, 207, 231
	14	4117.926	28, 46, 62, 78, 91, 105, 120, 135, 151, 166, 180, 193, 210, 232
	15	4120.045	28, 46, 62, 78, 91, 103, 116, 129, 143, 158, 172, 185, 197, 214, 235
	16	4121.861	28, 46, 62, 78, 91, 103, 116, 128, 142, 156, 170, 182, 193, 206, 223, 240

TABLE 6: Continued.

Image	n	Between-class variance (VTR)	Thresholds
Ostrich	2	1073.452	75, 135
	3	1139.260	69, 101, 149
	4	1178.650	65, 92, 125, 176
	5	1203.749	56, 78, 100, 131, 179
	6	1218.643	47, 65, 85, 103, 133, 181
	7	1228.925	47, 64, 83, 100, 122, 152, 192
	8	1236.023	45, 59, 75, 90, 104, 125, 155, 194
	9	1240.756	40, 52, 65, 80, 94, 107, 128, 157, 195
	10	1244.909	40, 52, 64, 78, 91, 103, 119, 141, 168, 201
	11	1247.879	40, 51, 62, 75, 87, 97, 108, 125, 148, 174, 205
	12	1250.313	37, 48, 57, 68, 80, 91, 101, 112, 128, 150, 175, 206
	13	1252.298	31, 44, 53, 63, 75, 86, 95, 105, 117, 134, 154, 178, 207
	14	1253.956	29, 42, 50, 58, 68, 79, 89, 98, 108, 120, 137, 157, 181, 209
	15	1255.366	29, 42, 50, 58, 67, 77, 86, 94, 102, 111, 124, 141, 160, 183, 210
	16	1256.490	28, 41, 49, 57, 66, 76, 85, 93, 101, 110, 122, 137, 155, 174, 196, 219
	Viaduct	2	7920.458
3		8117.991	54, 109, 193
4		8203.807	42, 84, 131, 203
5		8246.806	35, 68, 103, 146, 210
6		8272.775	31, 59, 88, 120, 160, 216
7		8287.714	28, 53, 77, 103, 132, 169, 220
8		8298.322	27, 51, 75, 100, 128, 164, 212, 246
9		8308.240	24, 45, 66, 88, 112, 139, 172, 216, 247
10		8315.133	22, 41, 60, 79, 99, 121, 146, 178, 219, 247
11		8320.032	20, 37, 54, 71, 89, 108, 128, 152, 183, 221, 248
12		8323.799	20, 36, 52, 68, 84, 101, 119, 139, 162, 190, 224, 248
13		8326.793	18, 33, 48, 62, 77, 92, 108, 125, 144, 167, 195, 226, 248
14		8329.119	17, 31, 44, 57, 70, 84, 98, 113, 129, 148, 170, 197, 227, 248
15		8330.983	17, 31, 44, 57, 70, 84, 98, 113, 129, 147, 169, 195, 224, 243, 251
16		8332.744	16, 29, 42, 54, 66, 78, 91, 104, 118, 133, 151, 172, 196, 224, 243, 251
Fish		2	3593.389
	3	3870.456	44, 104, 177
	4	3972.731	34, 81, 127, 188
	5	4024.885	27, 63, 101, 139, 194
	6	4054.836	24, 56, 90, 123, 156, 205
	7	4075.236	22, 49, 78, 107, 135, 168, 213
	8	4088.300	20, 44, 68, 93, 118, 142, 173, 216
	9	4097.101	19, 40, 62, 85, 107, 128, 149, 178, 218
	10	4103.194	18, 38, 58, 78, 98, 118, 137, 158, 185, 222
	11	4107.954	16, 34, 51, 69, 88, 107, 125, 143, 163, 190, 224
	12	4111.678	15, 31, 47, 64, 82, 100, 117, 133, 149, 169, 195, 227
	13	4114.783	14, 28, 43, 58, 74, 90, 106, 121, 136, 152, 172, 198, 228
	14	4116.954	14, 28, 43, 58, 73, 88, 103, 117, 131, 145, 160, 179, 203, 231
	15	4118.931	13, 25, 38, 51, 64, 78, 92, 106, 120, 133, 147, 162, 181, 205, 232
	16	4120.522	13, 25, 37, 49, 62, 75, 89, 102, 115, 127, 139, 152, 168, 188, 211, 235

TABLE 6: Continued.

Image	n	Between-class variance (VTR)	Thresholds
Houses	2	2543.788	56, 116
	3	2627.230	53, 105, 150
	4	2663.698	42, 69, 110, 152
	5	2694.902	39, 65, 96, 130, 158
	6	2708.655	39, 65, 95, 127, 150, 169
	7	2720.263	35, 55, 74, 98, 128, 151, 170
	8	2726.676	35, 55, 74, 98, 127, 148, 164, 184
	9	2732.853	32, 48, 65, 80, 101, 128, 148, 164, 184
	10	2736.510	30, 45, 60, 74, 87, 106, 129, 149, 164, 184
	11	2739.566	30, 45, 60, 73, 86, 105, 128, 145, 156, 168, 187
	12	2742.148	29, 42, 55, 68, 79, 94, 112, 130, 145, 156, 168, 187
	13	2744.071	28, 39, 51, 63, 74, 84, 98, 115, 131, 145, 156, 168, 187
	14	2745.542	28, 39, 51, 63, 74, 84, 98, 115, 131, 145, 155, 165, 176, 193
	15	2746.817	27, 37, 47, 58, 68, 77, 86, 100, 116, 131, 145, 155, 165, 176, 193
	16	2747.991	27, 37, 47, 57, 67, 76, 85, 98, 113, 127, 139, 147, 156, 165, 176, 193
	Mushroom	2	1988.328
3		2153.037	65, 110, 174
4		2237.441	59, 93, 135, 193
5		2277.427	52, 78, 106, 145, 199
6		2301.629	48, 71, 94, 122, 157, 205
7		2317.538	46, 67, 87, 110, 138, 171, 213
8		2328.526	44, 62, 79, 97, 118, 145, 177, 216
9		2335.812	42, 58, 74, 90, 107, 127, 152, 181, 218
10		2340.981	41, 56, 70, 84, 99, 116, 136, 159, 186, 220
11		2345.160	39, 52, 65, 78, 92, 107, 124, 144, 166, 191, 223
12		2348.413	38, 51, 63, 75, 87, 100, 115, 132, 152, 174, 199, 227
13		2350.988	38, 50, 62, 74, 85, 97, 110, 125, 142, 160, 180, 203, 229
14		2353.119	37, 48, 59, 69, 79, 89, 100, 113, 127, 144, 162, 181, 204, 230
15		2354.748	36, 46, 56, 66, 75, 84, 94, 105, 117, 130, 146, 163, 182, 205, 230
16		2356.113	36, 46, 56, 65, 74, 83, 92, 102, 113, 125, 139, 154, 169, 187, 208, 232
Snow mountain		2	1912.613
	3	2135.274	79, 137, 197
	4	2234.669	70, 119, 154, 204
	5	2288.799	55, 96, 129, 160, 206
	6	2317.224	52, 90, 118, 142, 167, 209
	7	2333.078	50, 85, 111, 132, 153, 174, 212
	8	2344.629	45, 76, 100, 120, 139, 158, 177, 214
	9	2352.951	44, 74, 97, 116, 133, 150, 167, 187, 220
	10	2359.175	41, 68, 90, 108, 124, 140, 156, 172, 193, 225
	11	2364.304	38, 62, 84, 102, 117, 132, 146, 160, 175, 196, 227
	12	2367.971	34, 56, 78, 96, 111, 125, 139, 152, 165, 179, 199, 228
	13	2371.208	33, 53, 73, 90, 104, 117, 130, 143, 155, 167, 180, 200, 229
	14	2373.567	32, 52, 72, 89, 103, 115, 127, 139, 150, 161, 172, 185, 205, 232
	15	2375.586	31, 49, 68, 84, 98, 109, 120, 131, 142, 153, 164, 175, 188, 208, 233
	16	2377.255	26, 42, 59, 75, 89, 101, 111, 121, 132, 143, 154, 165, 176, 189, 209, 234

TABLE 6: Continued.

Image	n	Between-class variance (VTR)	Thresholds
Snake	2	1118.615	87, 134
	3	1231.320	76, 114, 154
	4	1286.555	69, 101, 129, 166
	5	1317.027	63, 91, 115, 140, 175
	6	1336.172	59, 84, 105, 126, 149, 182
	7	1348.933	55, 78, 97, 115, 133, 155, 187
	8	1357.665	52, 73, 91, 107, 123, 140, 161, 192
	9	1364.159	50, 70, 87, 102, 116, 131, 148, 169, 198
	10	1368.955	47, 65, 81, 95, 108, 121, 135, 151, 172, 200
	11	1372.626	46, 63, 78, 91, 103, 115, 127, 140, 156, 176, 203
	12	1375.513	45, 61, 75, 88, 99, 110, 121, 133, 146, 162, 182, 208
	13	1377.847	43, 58, 72, 84, 95, 106, 116, 127, 138, 151, 166, 185, 211
	14	1379.736	42, 56, 69, 81, 92, 102, 112, 122, 132, 143, 155, 170, 189, 214
	15	1381.282	41, 54, 66, 77, 87, 97, 106, 115, 124, 134, 145, 157, 172, 191, 215
	16	1382.603	40, 52, 63, 74, 84, 93, 102, 111, 120, 129, 139, 150, 162, 177, 195, 219
	Riosanpablo	2	2667.020
3		2818.660	75, 121, 177
4		2892.439	68, 102, 143, 189
5		2931.654	62, 89, 121, 158, 197
6		2957.018	58, 82, 107, 138, 171, 204
7		2973.269	53, 74, 95, 119, 147, 177, 207
8		2984.972	50, 69, 87, 108, 133, 159, 185, 212
9		2993.359	48, 66, 83, 101, 122, 145, 168, 191, 215
10		2999.615	46, 63, 78, 93, 110, 130, 151, 173, 195, 217
11		3004.374	45, 61, 75, 89, 104, 121, 140, 160, 180, 200, 220
12		3008.082	44, 59, 72, 84, 97, 112, 129, 147, 166, 185, 203, 222
13		3011.101	43, 57, 69, 81, 93, 107, 122, 138, 155, 172, 189, 206, 224
14		3013.465	41, 54, 66, 77, 88, 100, 113, 128, 144, 160, 176, 192, 208, 225
15		3015.423	40, 53, 64, 74, 84, 95, 107, 121, 135, 150, 165, 180, 195, 210, 226
16		3017.081	39, 51, 62, 72, 82, 92, 103, 115, 128, 142, 156, 170, 184, 198, 212, 227

method [35], which is a local search method that cannot guarantee an optimal solution. Thus, its solutions are inferior to the solution produced by the algorithm using a global search.

5.2.3. Convergence Rate Comparison. The number of generations (NG) is a measure used for the convergence rate comparisons. If the target value, VTR, is achieved in a lesser number of generations (NG), it means a faster convergence rate for the algorithm. Table 3 shows the average of \overline{NG} (\overline{NG}_{AM}) for each specific number of thresholds. The results of each algorithm are represented in the corresponding column's name. In each column, the cell containing \overline{NG}_{AM} starts from the row associated with $n = 2$ until the row associated with $n = n_{max}$. The cells associated with $n = n_{max} + 1$ to the row

associated with $n = 16$ are filled by $NA(x)$. The second last row of the table is filled by the triple:

$$\left(n_{max}, NA \text{ (number of unsuccessful subproblem)}, \right. \\ \left. AM \left(\overline{NG}_{AM} \text{ of } n = 2 \text{ until } n = n_{max} \right) \right). \quad (23)$$

The algorithm with the highest n_{max} , that is, the lowest number of unsuccessful subproblems and the lowest average of generation is the winner. The ranking of the algorithms depends on the ordering of (a_1, b_1, c_1) and (a_2, b_2, c_2) as follows.

First, rank on a_1 and a_2 . Since both a_1 and a_2 are numeric, the higher value has the higher rank.

Second, rank on b_1 and b_2 . If a_1 is 16, then b_1 must be $NA(0)$. If a_1 is less than 16, then b_1 must be NA

(number of unsuccessful subproblems) and b_2 has the same characteristics as b_1 . Perform the order of the two numeric values in reverse; the lower value has the higher rank.

Third, rank on c_1 and c_2 . They are numeric but the lower is better; perform the order of the two numeric values in reverse. The lower value has the higher rank.

When the ordering is finished, assign the numeric value of “1” to the object having the highest rank, assign the numeric value of “2” to the first runner up, and so on. These values are represented in the last row of the table.

If the row having $n = m$ must be ranked, it can be done in the same manner as above with some minor modifications. If m is less than or equal to n_{\max} , the values of the triple pair will be m , $NA(0)$, \overline{NG}_{AM} . If m is greater than n_{\max} , the values of the triple pair will be $(n_{\max}, NA(x), \infty)$. Thus the ranking can now be performed.

From the ranking results, see the second last row of Table 3; the convergence rate can be ranked from best to worst in the following order: $O(0.2)R-DE$, $O(0.1)R-DE$, $O(0.0)R-DE$, $O(rand)R-DE$, $O(0.3)R-DE$, $O(0.4)R-DE$, $O(0.5)R-DE$, $O(0.6)R-DE$, DE , $O(0.7)R-DE$, $O(0.8)R-DE$, $rank-DE$, $O(0.9)R-DE$, ABC , PSO , $DPSO$, $FODPSO$.

As can be seen from Table 3, the DE algorithm cannot complete the task when $n > 12$ and the $rank-DE$ algorithm cannot complete the task when $n > 9$. Thus, $rank-DE$ cannot compete with DE on searching for global multilevel thresholding.

In order for $O(\beta)R-DE$ to outperform DE then β must be in the range of $[0.1, 0.6]$ or set to $rndreal[0, 1)$.

5.2.4. Stability Analysis. The harmonic mean of the success rate (SR_{HM}) for each specific number of thresholds was computed and is presented in Table 4. The results of each algorithm are represented in the corresponding column's name. The second row from the bottom shows the harmonic mean of the success rate of each algorithm for all threshold levels.

In each column, the cells containing SR_{HM} start from the row associated with $n = 2$ to the row associated with $n = n_{0.5}$. The cells from the row associated with $n = n_{0.5} + 1$ to the row associated with $n = n_{\max}$ are filled by SR_{HM} . The cells from the row associated with $n = n_{\max} + 1$ to the row associated with $n = 16$ are the cells that have SR_{HM} ; these cells will be excluded from the comparison. The second last row of the table is filled with the triple:

$$(n_{0.5}, n_{\max}, HM(SR_{HM} \text{ of } n = 2 \text{ until } n = n_{\max})). \quad (24)$$

The algorithm with the highest $n_{0.5}$, the highest n_{\max} , and the highest average success rate is the winner. The ranking of the algorithms depends on the ordering of (a_1, b_1, c_1) and (a_2, b_2, c_2) as follows.

First, rank on a_1 and a_2 .

Second, rank on b_1 and b_2 .

Third, rank on c_1 and c_2 .

Because they are numeric the higher value has the higher rank. When the ordering is finished, the numeric value of “1” is assigned to the object having the highest rank, the numeric value of “2” is assigned to the first runner up, and so on.

If the row having $n = m$ must be ranked, it can be done in the same manner as above with some minor modifications. If m is less than or equal to $n_{0.5}$, the values of the triple pair will be (m, m, SR_{HM}) . If $n_{0.5} < m \leq n_{\max}$, then the values of the triple pair will be $(n_{0.5}, m, SR_{HM})$. If m is greater than n_{\max} , the values of the triple pair will be $(n_{0.5}, n_{\max}, SR_{HM})$. Thus, the ranking can now be performed.

From the ranking results, see Table 4, the success rate can be ranked from best to worst in the following order: $O(0.0)R-DE$, $O(0.1)R-DE$, $O(0.2)R-DE$, $O(rand)R-DE$, $O(0.3)R-DE$, $O(0.5)R-DE$, $O(0.4)R-DE$, DE , $O(0.6)R-DE$, $O(0.7)R-DE$, $O(0.8)R-DE$, $rank-DE$, $O(0.9)R-DE$, ABC , PSO , $DPSO$, $FODPSO$, $multithresh$.

As can be seen from Table 4, the DE algorithm has an $SR_{HM} \geq 0.5$ until $n = 9$ and the $rank-DE$ algorithm has an $SR_{HM} \geq 0.5$ until $n = 7$. This result confirms that $rank-DE$ cannot compete with DE on searching for global multilevel thresholding. If the correct β is selected, the proposed algorithm can work very well. For $\beta \leq 0.5$, $O(\beta)R-DE$ has a higher rank than DE . The $multithresh$ function cannot compete with any of the other algorithms. It can also be seen that the proposed $O(0.0)R-DE$ algorithm has the best stability because its SR_{HM} is greater than 0.5 when $n = 2$ to 16.

5.2.5. Reliability Comparison. The threshold value distortion a.k.a. TVD for each specific threshold is computed, shown in Table 5 and depicted in Figure 2. The results of each algorithm are illustrated in the corresponding column's name. The second last row of Table 5 shows the slope or the approximated growth rate, δ , of the TVD of each algorithm for all threshold levels. The δ is the slope of the robust linear regression computed by the Matlab function “robustfit.” The lower slope exhibits the better reliability. The δ of each algorithm is sorted in descending order. From the results, the reliability can be ranked from best to worst in the following order: $O(0.0)R-DE$, $O(0.1)R-DE$, $O(0.2)R-DE$, $O(0.3)R-DE$, $O(0.4)R-DE$, $O(0.5)R-DE$, $O(rand)R-DE$, DE , $O(0.6)R-DE$, ABC , $rank-DE$, $O(0.7)R-DE$, $O(0.8)R-DE$, $O(0.9)R-DE$, $DPSO$, $FODPSO$, PSO , $multithresh$.

We can see from these results that $rank-DE$ has a higher approximated TVD growth rate than DE . $O(\beta)R-DE$ with $\beta \leq 0.5$ and $O(rand)R-DE$ are still better than DE . $O(0.0)R-DE$ produced the best result with a very flat slope and a very low y -intercept. The $multithresh$ function yielded a higher growth rate of solution distortion and a higher y -intercept than the other algorithms. The higher growth rate of solution distortion means the quality of solution drops very fast if the number of thresholds increases. The higher y -intercept means that the solution distortion at the lowest number of thresholds is high. Thus, the global optimization algorithm is required for solving the multilevel thresholding.

6. Conclusions

The differential evolution with onlooker (β) ranking-based mutation operator $O(\beta)R$ -DE algorithm was proposed and applied to the multilevel image thresholding problem. The objective of this proposed algorithm was to increase the ability for adjusting the balance of the exploitation and exploration abilities. Its concept is a combination of the ranking-difference evolution and onlooker selection of the ABC algorithm. The experiments compared the proposed $O(\beta)R$ -DE algorithm with six existing algorithms: PSO, DPSO, FODPSO, ABC, DE, and rank-DE on 20 real images of the Berkeley Segmentation Dataset and Benchmark and a satellite image. The stability analysis, convergence speed, and the reliability were measured. The results signified that the proposed $O(\beta)R$ -DE algorithm is more efficient than the six tested algorithms. The onlooker ranking-based mutation operator is able to enhance the performance of the proposed algorithm. The $O(\beta)R$ -DE not only obtained more stability analysis, but it also achieved faster convergence rates to reach the target BCV, if a proper value of β is set.

For future work based on this paper, the proposed $O(\beta)R$ -DE algorithm has one parameter to be set by a user; the mechanism to automatically adapt this parameter is not presented but is required.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the Higher Education Research Promotion and National Research University Project of Thailand, Office of the Higher Education Commission, through the Cluster of Research to Enhance the Quality of Basic Education.

References

- [1] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, New York, NY, USA, 1978.
- [2] J. S. Weszka, "A survey of threshold selection techniques," *Computer Graphics and Image Processing*, vol. 7, no. 2, pp. 259–265, 1978.
- [3] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, 1981.
- [4] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 233–260, 1988.
- [5] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.
- [6] A. T. Abak, U. Baris, and B. Sankur, "The performance evaluation of thresholding algorithms for optical character recognition," in *Proceedings of the 4th International Conference on Document Analysis and Recognition*, pp. 697–700, Ulm, Germany, August 1997.
- [7] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–168, 2004.
- [8] S. U. Lee and S. Yoon Chung, "A comparative performance study of several global thresholding techniques for segmentation," *Computer Vision, Graphics and Image Processing*, vol. 52, no. 2, pp. 171–190, 1990.
- [9] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Australia, December 1995.
- [11] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," Tech. Rep. TR-95-012, International Computer Sciences Institute, Berkeley, Calif, USA, 1995.
- [12] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [13] M. S. Couceiro, R. P. Rocha, N. M. F. Ferreira, and J. A. T. Machado, "Introducing the fractional-order Darwinian PSO," *Signal Image and Video Processing*, vol. 6, no. 3, pp. 343–350, 2012.
- [14] R. V. Kulkarni and G. K. Venayagamoorthy, "Bio-inspired algorithms for autonomous deployment and localization of sensor nodes," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 40, no. 6, pp. 663–675, 2010.
- [15] B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding," *Applied Soft Computing Journal*, vol. 3, no. 6, pp. 3066–3091, 2013.
- [16] K. Hammouche, M. Diaf, and P. Siarry, "A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 676–688, 2010.
- [17] V. Osuna-Enciso, E. Cuevas, and H. Sossa, "A comparison of nature inspired algorithms for multi-threshold image segmentation," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1213–1219, 2013.
- [18] P. Ghamisia, M. S. Couceiro, F. Martins, and J. A. Benediktsson, "Multilevel image segmentation based on Fractional-Order Darwinian particle swarm optimization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 1–44, 2013.
- [19] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2066–2081, 2013.
- [20] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer, Berlin, Germany, 2005.
- [21] R. Storn and K. Price, *Home Page of Differential Evolution*, International Computer Science Institute, Berkeley, Calif, USA, 2010.
- [22] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, pp. 1785–1791, September 2005.
- [23] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference*, pp. 485–492, July 2006.

- [24] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [25] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 38, no. 1, pp. 218–237, 2008.
- [26] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [27] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 107–125, 2008.
- [28] R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.
- [29] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th International Conference on Computer Vision*, vol. 2, pp. 416–423, July 2001.
- [30] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, and N. M. F. Ferreira, "An efficient method for segmentation of image based on fractional calculus and natural selection," *Expert Systems with Applications*, vol. 39, pp. 12407–12417, 2012.
- [31] R. Gamperle, S. D. Muller, and A. Koumoutsakos, "A parameter study for differential evolution," in *Proceedings of the Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, vol. 10, pp. 293–298, 2002.
- [32] M. Saraswat, K. V. Arya, and H. Sharma, "Leukocyte segmentation in tissue images using differential evolution algorithm," *Swarm and Evolutionary Computation*, vol. 11, pp. 46–54, 2013.
- [33] J. Kittler and J. Illingworth, "On threshold selection using clustering criteria," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 5, pp. 652–655, 1985.
- [34] R. Guo and S. M. Pandit, "Automatic threshold selection based on histogram modes and a discriminant criterion," *Machine Vision and Applications*, vol. 10, no. 5-6, pp. 331–338, 1998.
- [35] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

Research Article

Fuzzy-Based Segmentation for Variable Font-Sized Text Extraction from Images/Videos

Samabia Tehsin,¹ Asif Masood,¹ Sumaira Kausar,² and Fahim Arif¹

¹ MCS, National University of Science & Technology (NUST), Islamabad, Pakistan

² College of E & ME, National University of Science & Technology (NUST), Islamabad, Pakistan

Correspondence should be addressed to Samabia Tehsin; tsamabia@yahoo.com

Received 28 October 2013; Revised 23 December 2013; Accepted 14 January 2014; Published 19 March 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Samabia Tehsin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Textual information embedded in multimedia can provide a vital tool for indexing and retrieval. A lot of work is done in the field of text localization and detection because of its very fundamental importance. One of the biggest challenges of text detection is to deal with variation in font sizes and image resolution. This problem gets elevated due to the undersegmentation or oversegmentation of the regions in an image. The paper addresses this problem by proposing a solution using novel fuzzy-based method. This paper advocates postprocessing segmentation method that can solve the problem of variation in text sizes and image resolution. The methodology is tested on ICDAR 2011 Robust Reading Challenge dataset which amply proves the strength of the recommended method.

1. Introduction

Recently there has been a rapid surge in multimedia reservoirs that raised the need of efficient retrieval, indexing, and browsing of multimedia information. Several methodologies are presented in the literature to retrieve image and video data, which exploit color, texture, shape, and relation between objects, and so forth. However, embedded text in images can be extraordinarily instrumental for data retrieval as visual texts in multimedia communicate information regarding news headlines, title of movie, trade-name of products, summaries of sports contest, date and time of events, and so forth. Such information can be influential for the understanding and retrieval of images or videos.

Text implanted in images may be categorized in two classes, namely, caption text and scene text. Caption text is imposed over the image in the editing process for example news headings and match summary/score. It is also referred to as artificial text or superimposed text, whereas scene text is an actual part of the scene, that is, brand name of the product during commercial break, text on sign-board, name plate and text visible on dresses or product, and so forth.

One of the key challenges posed to the text detection process is to deal with text size variations. The text variation

may be classified in two types: firstly, the variation of spatial resolution of images and secondly the variation of font sizes within an image. This paper focuses on the above mentioned problem in text detection and provides viable solutions for both categories of the problem.

The rest of the paper is ordered as follows. Section 2 highlights some related work of the field. Section 3 introduces the proposed method to segment text in images. Section 4 presents the dataset used and results of text segmentation algorithm. Section 5 provides some concluding remarks.

2. Literature Review

A variety of techniques for text extraction have appeared in the recent past [1–6]. Comprehensive surveys can be traced explicitly in [7–9]. These techniques can be categorized into two types mainly with reference to the utilized text features, that is, region-based and texture-based methods [10]. Texture-based methods pertain to textural properties of the text, distinguishing it from the background. These techniques mostly use Gabor filters, Wavelet, Fast Fourier transform, Spatial variance, and so forth. This approach further uses machine learning methods such as support

vector machine (SVM), multilayer perceptron (MLP), and adaBoost [11–15]. Region-based methods use distinct region features to extract text content. This methodology deals with the color dissimilarity of the text and its surrounding pixels. Procedures based on color, edge, and connected components are frequently exercised in this category [16–19]. These techniques typically work in the bottom up fashion by initially segmenting the small regions and later grouping the potential text regions. Region-based methods are generally composed of three modules: (1) segmenting the image into small regions which aims at segregating the character regions from its background, (2) merging and grouping of small regions to form words and sentences, and (3) differentiating between text and nontext objects.

Segmentation identifies the occurrence of different regions in the image but does not recognize the relation between these regions. It is substantial to merge the characters of a word to form a text object, because most of the text detection techniques work on group of characters and it is very difficult to detect the isolated character [20, 21]. This grouping can utilize the pixel level features or can exploit the high level features.

Presently, few pixel level merging methods are introduced in the literature pertaining to text detection. Dilation is the most commonly used merging technique [22–26], wherein the dimensions of the morphological operator intrinsically characterize the range of the homogeneous segmented regions. Consequently, hefty text blocks are tending to oversegmentation, whereas diminutive text areas are possibly skipped. Fixed size of the structuring element can only materialize for limited spatial resolution and small range of font sizes. Besides, size of the structuring element should be dependent upon the size of the text but usually has the fixed value which cannot deal with the variation in resolution of image and size of text. Some methodologies in literature utilize pyramid approach to solve this problem and extend the range of text sizes for detection [23, 27, 28]. This highly increases the computational requirements or demands for parallel processing mechanisms.

Object level merging is more close to human vision and deals with the objects and regions instead of pixels. It connects the potential character objects to form the text strings. Hence, the grouping and merging are dependent upon some high level features which gives better performance.

Wolf and Jolion [29] used disparity in heights and positions of the connected component to merge the characters. Minetto et al. [27] developed a grouping step, based on the space between the two text areas relative to their height. Pan et al. [30] built component relation using minimum spanning tree. This text detection method merges the characters into words using shape and spatial difference. Gonzalez and Bergasa [31] suggested that characters of the same word should have several similar characteristics, for instance, stroke size, altitude, position, adjacency, and constant inter-letter and interword spacing.

Shi et al. [32] used the graph model to merge the neighboring regions to form text strings. The adjoining nodes for each node are those ones that persuade the certain conditions based upon difference in color, position, width

ratio, and height ratio. Character candidates are linked into pairs in Yao et al. [33] method. If two regions have similar stroke widths (ratio between the mean stroke widths is fewer than 2.0), matching sizes (ratio between their characteristic scales does not surpass 2.5), and similar colors and are closely placed (distance between them is less than two times the sum of their characteristic scales), they are tagged as a couple. Subsequently, a greedy hierarchical agglomerative clustering approach is exercised to combine the pairs into candidate chains.

Though these features are defined by strict boundaries in the existing techniques, the relation between the neighboring characters is not crisp. It is principally inequitable to declare a character as a neighbor if its distance to height ratio is 1.50 or less, whereas the same verdict gets void, if the ratio turns to even 1.51. The parameters to define the proximity of potential character should have been diffused instead of crisp logic. Thus, there is a need to architect a merging process in which the rules of inference are formulated in a general way, utilizing diffused categories. There is a requirement to frame a system which gives some weight to each of the features used for measuring the degree of neighborhood. Moreover, the similarity obtained by the currently reported features mostly does not correspond to human perception. Human perception of propinquity, similar heights, and similar color cannot be fully expressed using discrete and rigid boundaries or thresholds. These linguistic variables can be better defined by the fuzzy logic.

3. Methodology

Component extraction or segmentation is the procedure of dividing a digital image into multiple fragments, called super-pixels [34, 35]. The objective of segmentation is to reduce the computational complexity of the under process image and make its representation easier to analyze. Image segmentation is classically used to trace objects and boundaries in images. In particular, image segmentation is the process to label the pixels of image, where the pixels with same labels share some common characteristics such as color, intensity, and texture and; moreover, edge detection is a basic instrument used in most image processing applications to obtain sharp alteration in intensity of the region boundaries.

Proposed segmentation method consists of two processes: splitting and merging. Splitting is performed by the traditional region-based segmentation techniques, whereas merging is based on the novel fuzzy-based method. Figure 1 provides the architecture of the proposed work.

3.1. Splitting. There exists sharp transition between the text and its background. Edge detection is the budding segmentation tool for text images because sharp intensity transition is the common feature in all the text objects. Exploiting this feature, edge detection along with the connected component labeling is used for segmentation in the proposed methodology, where Sobel edge detection technique is used for edge detection, and image dilation is applied to connect the broken edges.

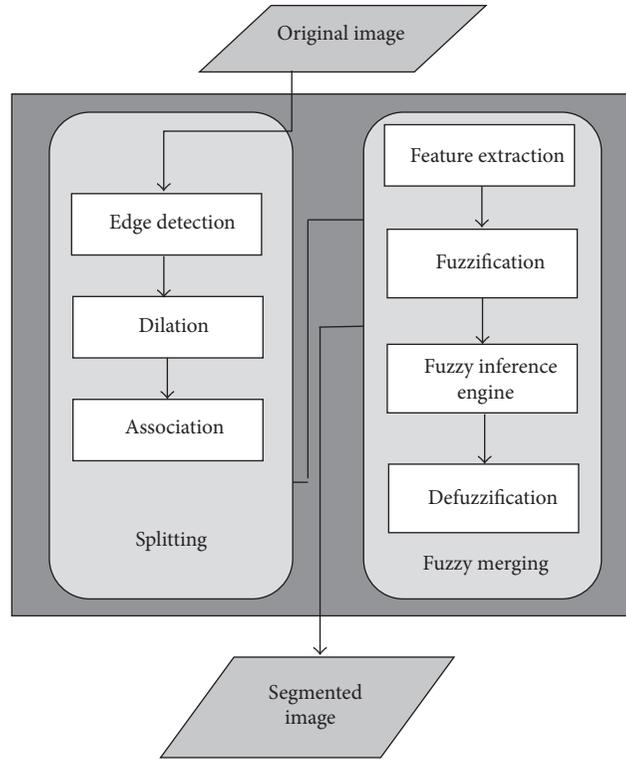


FIGURE 1: Architecture of the proposed method.

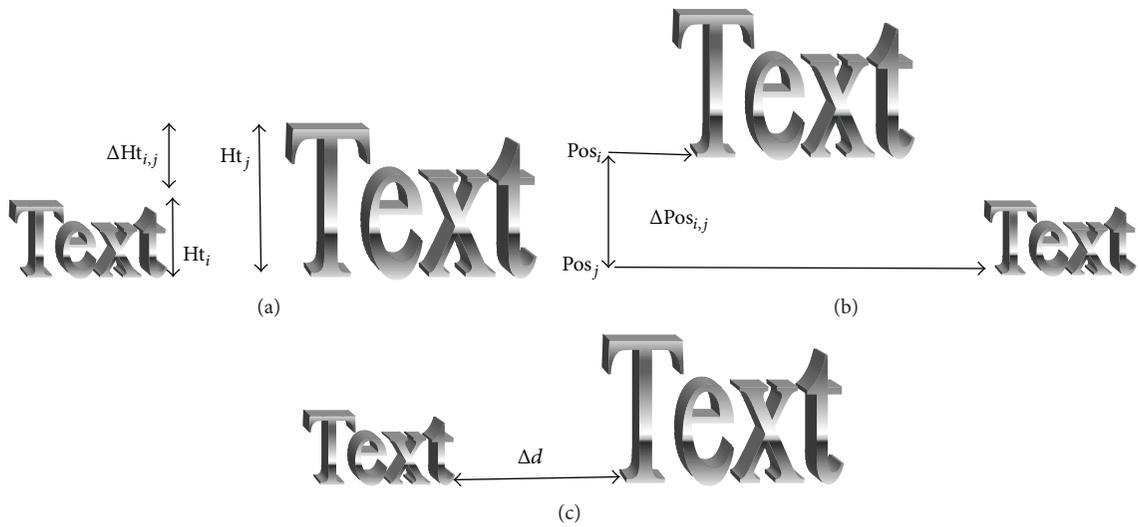


FIGURE 2: Factors fed into the fuzzy system: (a) height, (b) position, and (c) distance.

Adaptive size of the dilation operator is calculated in consonance with the resolution of the image, which ranges between 3 and 5% of the width of the image. Dilation is performed prior to the fuzzy merging just to minimize the computational efforts. Proposed fuzzy merging method can work without this morphological operation.

3.2. *Fuzzy Merging.* Succeeding section explains the fuzzy merging process.

Let I be the input image and R the set of all the regions of I , extracted by the above mentioned method. Let $I = \{R_1, R_2, R_3, \dots, R_m\}$, and m is the total number of regions in the image I .

The problem of merging process can be defined using the graph theory. Let G denote the undirected graph and $V(G) = R$ represent the vertices of the graph G . Edges of the graph G are $E(G) = \{(R_i, R_j) \in R \times R \mid i \neq j\}$. These edges show the probability of joining two vertices. Initially, $\forall e \in E(G)$ are set

to null. This probability $p_{i,j}$ can be calculated by fuzzy logic and based upon the four factors. Four factors considered are explained later in the paper.

Fuzzy-based methods assign gradual membership value to the objects, to join with other text instances, which are measured as degrees in $[0, 1]$. This gives the flexibility to connect the object based on more than one feature, depending upon the different membership values of all the parameters.

3.2.1. Feature Extraction. The merging of character candidates relies on number of factors. Four features are extracted for the decision of joining objects as words or sentences. These features are color, height, position, and distance.

Color. Color is taken as the parameter to join the two text objects. Color of the characters of a single word or sentence is mostly the same. If the colors of two text objects are similar, then these objects can be the candidates to merge. In order to get the degree of similarity, difference between the two colors is calculated.

Lab color coding is used to describe the color of the object. Unlike RGB and CMYK, Lab color coding approximates the human vision system. ΔE is described in the $L^*C^*h^*$ color space with differences in lightness, chroma, and hue calculated from $L^*a^*b^*$ coordinates. Difference of two colors having coordinates (L_1^*, a_1^*, b_1^*) and (L_2^*, a_2^*, b_2^*) can be defined as

$$\Delta E = \sqrt{\left(\frac{\Delta L^*}{K_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{K_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{K_H S_H}\right)^2}. \quad (1)$$

Here,

$$\begin{aligned} \Delta L^* &= L_1^* - L_2^*, \\ C_1^* &= \sqrt{a_1^{*2} + b_1^{*2}}, \\ C_2^* &= \sqrt{a_2^{*2} + b_2^{*2}}, \\ \Delta C_{ab}^* &= C_1^* - C_2^*, \\ \Delta H_{ab}^* &= \sqrt{\Delta E_{ab}^{*2} - \Delta L^{*2} - \Delta C_{ab}^{*2}} \\ &= \sqrt{\Delta a^{*2} + \Delta b^{*2} + \Delta C_{ab}^{*2}}, \\ \Delta a^* &= a_1^* - a_2^*, \\ \Delta b^* &= b_1^* - b_2^*, \\ S_L &= 1, \\ S_C &= 1 + K_1 C_1^*, \\ S_H &= 1 + K_2 C_1^*, \end{aligned}$$

$$K_L = 1,$$

$$K_1 = 0.045,$$

$$K_2 = 0.015. \quad (2)$$

Geometrically, the amount ΔH_{ab}^* presents the arithmetic mean of the chord lengths of the equal chroma circles of the two colors.

Height. Difference of heights is the second input parameter for fuzzy system. Only objects with similar heights should be merged because characters of the same word or sentence have the same or similar heights. Difference of heights of two objects is measured as follows:

$$\Delta H_{t_i,j} = \frac{|H_{t_i} - H_{t_j}|}{H_{t_i}}, \quad (3)$$

where H_{t_i} and H_{t_j} are the heights of i th and j th objects, respectively.

Position. Position of the two objects should be the same for merger. This merging process is proposed for horizontal text only as most of the caption text is horizontally aligned. This can be expanded to other directions by considering position at different angles. Consider

$$\Delta \text{Pos}_{i,j} = \frac{|\text{Pos}_i - \text{Pos}_j|}{H_{t_i}}, \quad (4)$$

where Pos_i and Pos_j are the bottom coordinates of bounding boxes of i th and j th objects, respectively.

Distance. Characters of the same word or sentence are placed closely. The distance between characters varies with the variation in font size and is highly dependent upon the heights of the characters. Distance (Δd) between two regions in an image is calculated by

$$\Delta d = \frac{\min(|x_i(1) - x_j(2)|, |x_j(1) - x_i(2)|)}{H_{t_i}}, \quad (5)$$

where $x_n(1)$ and $x_n(2)$ are the left and right coordinates of bounding box of n th object. Figure 2 explains the height, position, and distance phenomena pictorially.

3.2.2. Fuzzification. This step gets the inputs and decides the degree to which suitable fuzzy sets belong by means of membership functions. The input has to be a crisp numerical value bounded to the universe of discourse of the input variable and the output is a fuzzy degree of membership in the qualifying linguistic set. Fuzzification of the input refers to either a table lookup or function estimation.

Let the inputs to the fuzzy system be represented in the vector notation:

$$\begin{aligned} x^* &= [x_1^* \ x_2^* \ x_3^* \ x_4^*] \\ &= [\Delta E \ \Delta H_t \ \Delta \text{Pos} \ \Delta d], \end{aligned} \quad (6)$$

where x^* belonging to R^4 represents real value points. We define symmetric Gaussian function and sigmoid function for the input.

The symmetric Gaussian function is defined by two parameters σ and c :

$$\begin{aligned}\mu_{A^e}(x_1) &= e^{-\frac{1}{2}\left(\frac{x_1 - \bar{x}_1^{(e)}}{\sigma_1^{(e)}}\right)^2}, \\ \mu_{B^f}(x_2) &= e^{-\frac{1}{2}\left(\frac{x_2 - \bar{x}_2^{(f)}}{\sigma_2^{(f)}}\right)^2}, \\ \mu_{C^g}(x_3) &= e^{-\frac{1}{2}\left(\frac{x_3 - \bar{x}_3^{(g)}}{\sigma_3^{(g)}}\right)^2}, \\ \mu_{D^h}(x_4) &= e^{-\frac{1}{2}\left(\frac{x_4 - \bar{x}_4^{(h)}}{\sigma_4^{(h)}}\right)^2},\end{aligned}\quad (7)$$

where $e = 1, 2$; $f = 1, 2$; $g = 1, 2$; and $h = 1, 2$ represent the number of fuzzy sets. $\bar{x}_1^{(e)}$, $\bar{x}_2^{(f)}$, $\bar{x}_3^{(g)}$, and $\bar{x}_4^{(h)}$ represent the means of fuzzy sets, where $\sigma_1^{(e)}$, $\sigma_2^{(f)}$, $\sigma_3^{(g)}$, and $\sigma_4^{(h)}$ represent the variances of fuzzy sets.

Third membership function of all the inputs exhibits a progression from miniature start that advances and reached a culmination over time. Sigmoid function is used to express this phenomenon. Consider

$$\begin{aligned}\mu_{A^s}(x_1) &= \frac{1}{1 + e^{-a_s(x_1 - c_s)}}, \\ \mu_{B^t}(x_2) &= \frac{1}{1 + e^{-a_t(x_2 - c_t)}}, \\ \mu_{C^u}(x_3) &= \frac{1}{1 + e^{-a_u(x_3 - c_u)}}, \\ \mu_{D^v}(x_4) &= \frac{1}{1 + e^{-a_v(x_4 - c_v)}},\end{aligned}\quad (8)$$

where $s = 3$; $t = 3$; $u = 3$; and $v = 3$ represent the fuzzy set's number. $a_{s...v}$ and $c_{s...v}$ are the model parameters to be fitted.

The following function is used to map x^* belonging to R^4 into fuzzy set $ABCD$:

$$\begin{aligned}\mu_{ABCD}(x_1, x_2, x_3, x_4) \\ &= \mu_A(x_1) * \mu_B(x_2) * \mu_C(x_3) * \mu_D(x_4) \\ &= \min(\mu_A(x_1), \mu_B(x_2), \mu_C(x_3), \mu_D(x_4)).\end{aligned}\quad (9)$$

Minimum t -norm operator ($*$) is used for fuzzification.

3.2.3. Product Inference Engine. Multiple inputs and single output fuzzy rule-base is employed for the current merging problem. Product inference engine (PIE) makes use of fuzzy rule base and linguistic rules. PIE encompasses individual

rule-based inference with union combination, min implication, min operator for t -norm, and max operator for s -norm:

$$\begin{aligned}\text{Ru}^{(1)} : & \text{IF } \Delta E \text{ is } A^1 \text{ and } \Delta Ht \text{ is } B^1 \\ & \text{and } \Delta Pos \text{ is } C^1 \text{ and } \Delta d \text{ is } D^1 \text{ THEN } y \text{ is } Z^2; \\ \text{Ru}^{(2)} : & \text{IF } \Delta E \text{ is } A^2 \text{ and } \Delta Ht \text{ is } B^1 \\ & \text{and } \Delta Pos \text{ is } C^1 \text{ and } \Delta d \text{ is } D^1 \text{ THEN } y \text{ is } Z^2; \\ \text{Ru}^{(3)} : & \text{IF } \Delta E \text{ is } A^3 \text{ or } \Delta Ht \text{ is } B^3 \\ & \text{or } \Delta Pos \text{ is } C^3 \text{ or } \Delta d \text{ is } D^3 \text{ THEN } y \text{ is } Z^1; \\ \text{Ru}^{(4)} : & \text{IF } \Delta E \text{ is } A^1 \text{ and } \Delta Ht \text{ is } B^2 \\ & \text{and } \Delta Pos \text{ is } C^1 \text{ and } \Delta d \text{ is } D^1 \text{ THEN } y \text{ is } Z^2; \\ \text{Ru}^{(5)} : & \text{IF } \Delta E \text{ is } A^1 \text{ and } \Delta Ht \text{ is } B^2 \\ & \text{and } \Delta Pos \text{ is } C^2 \text{ and } \Delta d \text{ is } D^1 \text{ THEN } y \text{ is } Z^2; \\ \text{Ru}^{(6)} : & \text{IF } \Delta E \text{ is } A^1 \text{ and } \Delta Ht \text{ is } B^1 \\ & \text{and } \Delta Pos \text{ is } C^2 \text{ and } \Delta d \text{ is } D^1 \text{ THEN } y \text{ is } Z^2; \\ \text{Ru}^{(7)} : & \text{IF } \Delta E \text{ is } A^1 \text{ and } \Delta Ht \text{ is } B^1 \\ & \text{and } \Delta Pos \text{ is } C^1 \text{ and } \Delta d \text{ is } D^2 \text{ THEN } y \text{ is } Z^2,\end{aligned}\quad (10)$$

where (A^1, A^2, A^3) , (B^1, B^2, B^3) , and (C^1, C^2, C^3) are the input fuzzy membership functions for the same, similar, and different, (D^1, D^2, D^3) are the membership values for minimum, average, and maximum and (Z^1, Z^2) are the output membership functions corresponding to *Not join* and *join*. Triangular curve function is used as output membership function that can be defined by

$$\mu_Z(y_m) = \begin{cases} 0 & y_m \leq a_z \text{ OR } c_z \leq y_m \\ \frac{y_m - a_z}{b_z - a_z} & a_z \leq y_m \leq b_z \\ \frac{c_z - y_m}{c - b_z} & b_z \leq y_m \leq c_z. \end{cases}\quad (11)$$

More compactly, it can be expressed as

$$\mu_Z(y_m) = \max\left(\min\left(\frac{y_m - a_z}{b_z - a_z}, \frac{c_z - y_m}{c_z - b_z}\right), 0\right).\quad (12)$$

The parameters a_z and c_z define the feet of the triangle and the parameter b_z defines the peak. Figure 3 shows different membership functions used in the system.

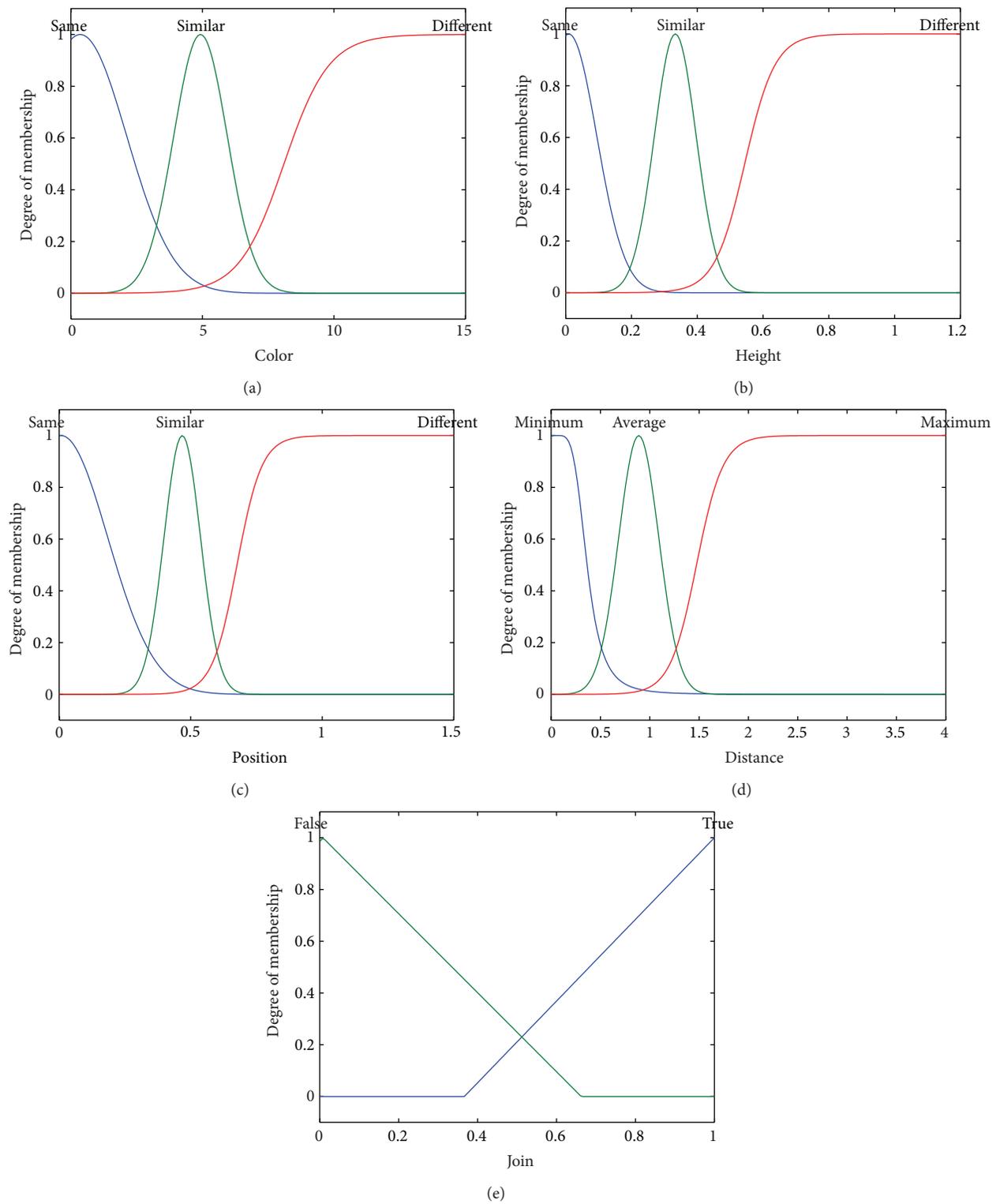


FIGURE 3: Membership functions: (a) color, (b) height, (c) position, (d) distance, and (e) output.

PIE can be fully defined by

$$\begin{aligned} & \mu_{Z'}(y_m) \\ &= \max_{\{e,f,g,h,s,t,u,v,z\}} \left\{ \sup_{x \in X} \min \left(\mu_{ABCD}(x_1, x_2, x_3, x_4), \right. \right. \\ & \quad \left. \left. \mu_{A'}(x_1), \mu_{B'}(x_2), \mu_{C'}(x_3), \right. \right. \\ & \quad \left. \left. \mu_{D'}(x_4), \mu_Z(y_m) \right) \right\}. \end{aligned} \quad (13)$$

3.2.4. Defuzzification. Defuzzification is the mapping of fuzzy values into the real-world values. Center average fuzzifier (CAD) is used as the weighted average of the centers of fuzzy sets as it provides a reasonable approximation:

$$\bar{\omega}_m = y_m^* = \frac{\sum_{c=1}^{n_r} \bar{y}_n^{(c)} \omega_n^{(c)}}{\sum_{c=1}^{n_r} \omega_n^{(c)}}, \quad (14)$$

where $\bar{y}_n^{(c)}$ and $\omega_n^{(c)}$ are the center and height of the output fuzzy sets. CAD is chosen because it is computationally less expensive and has more accuracy and continuity when compared to other defuzzifiers [36].

4. Results and Experiments

Dataset of ICDAR 2011 Robust Reading Competition, Challenge 1: “Reading Text in Born-Digital Images (Web and Email),” is applied in this research, wherein the dataset comprises 102 images of test and 420 images of training sets. The above dataset possesses vast variation in font size, resolution, background complexity, and font type. However, ICDAR dataset is recognized as the most widely used benchmark for text detection.

The ranking metric used for the text segmentation task is accurate. Accuracy of segmentation can be defined as

$$\text{Acc} = \sum_{i=1}^N \frac{\text{No. of correctly segmented objects}}{\text{Total number of objects}}. \quad (15)$$

In the text detection and localization problem, isolated character is also considered as under segmentation. Proposed method obtained 90.7% accuracy for segmentation of text objects. Comparison of the segmentation results with and without fuzzy merging can be viewed in Figure 4. Segmentation without fuzzy merging is tested for adaptive and fixed size structuring elements. Achieved results show that fuzzy merging has a very effective role in segmentation for text detection.

In order to prove the practicability of the proposed segmentation method, fuzzy merging is added as the post segmentation process in textorter [37], which is the best technique in ICDAR Robust Reading Competition 2011 [38], whereby the results justify a major improvement in the detection rate of textorter. It is also factual that many isolated characters are not detected as text by textorter, as these are not merged as a complete word. The ranking metric used

TABLE 1: Comparison of proposed work with other techniques.

Method	Recall	Precision	Harmonic mean
Textorter with fuzzy merging	73.75	85.12	79.02
Textorter [37]	69.62	85.83	76.88
TH-TextLoc*	73.08	80.51	76.62

*Stood second in ICDAR Robust Reading Competition 2011 [38].

for the text localization task is the harmonic mean which is computed according to the methodology proposed in [39]. It is a combination of two measures: precision and recall. Table 1 shows the comparison of results for different text detection methods.

Figure 5 shows the superiority of the proposed method. Results show that fuzzy merging really enhances the segmentation process for text detection.

Different combinations for input and output membership functions are tested, where the results show that the combination testified in the proposed methodology ensures the best outcome. Gaussian, triangular, sigmoid, trapezoidal, and bell-shaped are commonly used membership functions. These functions are tested for making different combinations of fuzzy inference engine. Gaussian, triangular, and sigmoid functions are defined in Section 3.2.2. Bell-shaped function can be defined as

$$\mu_A(x) = \frac{1}{1 + |(x - c)/a|^{2b}}. \quad (16)$$

The comprehensive bell function can be defined using three parameters a , b , and c , and here the parameter b is mostly positive, whereas parameter c traces the middle of the curve.

The trapezoidal curve is a function of a vector “ X ” and is dependent upon four scalar parameters a , b , c , and d :

$$\mu_A(x) = \begin{cases} 0, & x < a \\ \frac{x - a}{b - a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d - x}{d - c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (17)$$

or it can be defined compactly as

$$\mu_A(x) = \max \left(\min \left(\frac{x - a}{b - a}, 1, \frac{d - x}{d - c} \right), 0 \right). \quad (18)$$

The parameters a and d trace the “feet” of the trapezoid and the parameters b and c set the “shoulders.”

Figure 6 shows the comparison of different membership functions regarding four inputs.

5. Conclusion

The paper addresses very crucial problem of text detection, which is variation in font size and resolution. Earlier approaches are primarily dataset specific and unable to deal

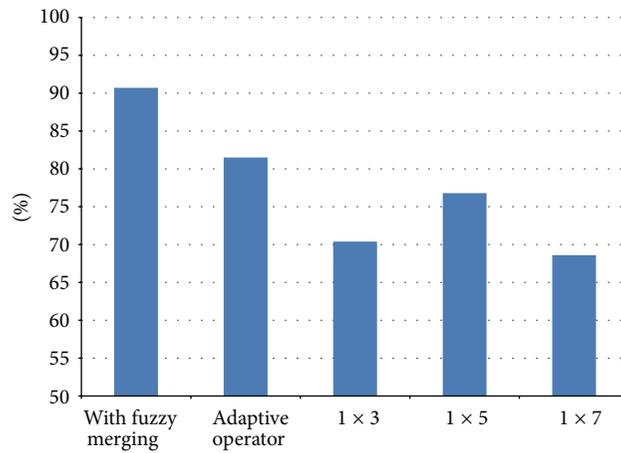


FIGURE 4: Comparison of proposed methodology with other techniques.



FIGURE 5: Results of the proposed method. (a) Original images, (b) after splitting method, and (c) after fuzzy merging.

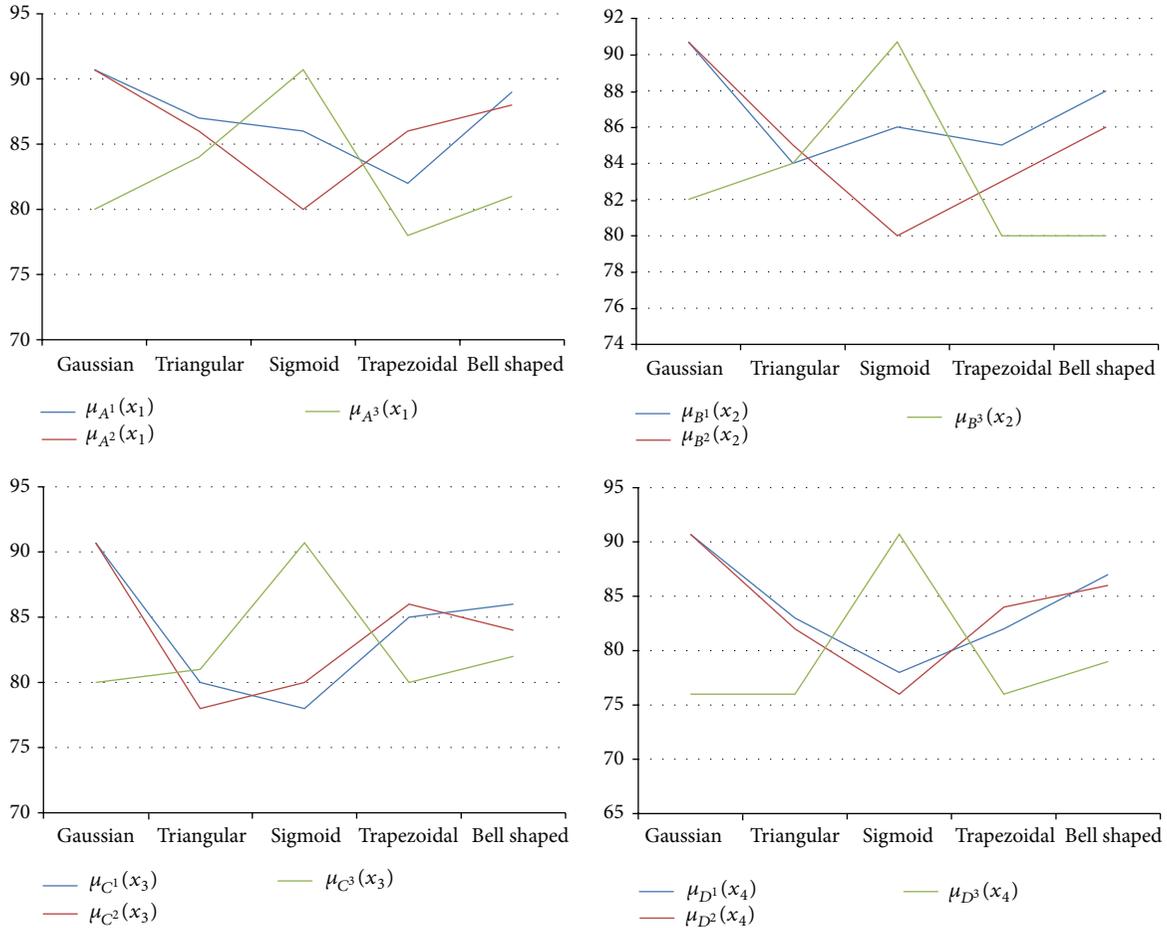


FIGURE 6: Comparison of different membership functions.

with enormous variation of font sizes. This paper devises a fuzzy-based postprocessing method for segmentation duly operatable with combination of any segmentation method. Four factors are mainly put forth for joining characters into words. These factors are fed into the fuzzy system which gives the verdict of joining or not joining regions. Dataset of ICDAR 2011 Robust Reading Competition, Challenge 1: “Reading Text in Born-Digital Images (Web and Email),” is applied into this research, whereby the results achieved stand out to be productive when pitched against the above referred retrieval problems.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

References

[1] H. Li, D. Doermann, and O. Kia, “Automatic text detection and tracking in digital video,” *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 147–156, 2000.

[2] K. I. Kim, K. Jung, and J. H. Kim, “Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1631–1639, 2003.

[3] M. Zhao, S. Li, and J. Kwok, “Text detection in images using sparse representation with discriminative dictionaries,” *Image and Vision Computing*, vol. 28, no. 12, pp. 1590–1599, 2010.

[4] K. Wang and S. Belongie, “Word spotting in the wild,” in *Proceedings of the European Conference on Computer Vision (ECCV ’10)*, pp. 591–604, Springer, 2010.

[5] L. Neumann and J. Matas, “A method for text localization and recognition in real-world images,” in *Proceedings of the Asian Conference on Computer Vision (ACCV ’11)*, pp. 770–783, Springer, 2011.

[6] P. Shivakumara, T. Q. Phan, and C. L. Tan, “A Laplacian approach to multi-oriented text detection in video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 412–419, 2011.

[7] K. Jung, K. I. Kim, and A. K. Jain, “Text information extraction in images and video: a survey,” *Pattern Recognition*, vol. 37, no. 5, pp. 977–997, 2004.

[8] J. Liang, D. Doermann, and H. Li, “Camera-based analysis of text and documents: a survey,” *International Journal on*

- Document Analysis and Recognition*, vol. 7, no. 2-3, pp. 84–104, 2005.
- [9] C. P. Sumathi, T. Santhanam, and G. Gayathri, “A Survey on various approaches of text extraction in images,” *International Journal of Computer Science & Engineering Survey*, vol. 3, no. 4, 2012.
- [10] R. Lienhart, *Video OCR: A Survey and Practitioner’s Guide*, Video mining, Springer, Burlingame, Calif, USA, 2003.
- [11] C. Li, X. G. Ding, and Y. S. Wu, “An algorithm for text location in images based on histogram features and Ada-boost,” *Journal of Image and Graphics*, vol. 3, article 003, 2006.
- [12] K. I. Kim, K. Jung, and J. H. Kim, “Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1631–1639, 2003.
- [13] R. Lienhart and A. Wernicke, “Localizing and segmenting text in images and videos,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 256–268, 2002.
- [14] J. Gllavata, E. Qeli, and B. Freisleben, “Detecting text in videos using fuzzy clustering ensembles,” in *Proceedings of the 8th IEEE International Symposium on Multimedia (ISM ’06)*, pp. 283–290, IEEE, December 2006.
- [15] D. Chen, J.-M. Odobez, and H. Bourlard, “Text detection and recognition in images and video frames,” *Pattern Recognition*, vol. 37, no. 3, pp. 595–608, 2004.
- [16] J. Fabrizio, M. Cord, and B. Marcotegui, *Text Extraction from Street Level Images, City Models, Roads and Traffic (CMRT)*, 3, 2009.
- [17] M. León Cristóbal, V. Vilaplana Besler, A. Gasull Llampallas, and F. Marqués Acosta, *Region-Based Caption Text Extraction*, 2012.
- [18] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’10)*, pp. 2963–2970, IEEE, June 2010.
- [19] M. Anthimopoulos, B. Gatos, and I. Pratikakis, “A two-stage scheme for text detection in video images,” *Image and Vision Computing*, vol. 28, no. 9, pp. 1413–1426, 2010.
- [20] L. Neumann and J. Matas, “A method for text localization and recognition in real-world images,” in *Proceedings of the Asian Conference on Computer Vision (ACCV ’10)*, pp. 770–783, Springer, 2011.
- [21] S. T. Deepa and S. P. Victor, “A novel method for text extraction,” *International Journal of Engineering Science & Advanced Technology*, no. 4, pp. 961–964, 2013.
- [22] R. Farhoodi and S. Kasaei, “Text segmentation from images with textured and colored background,” in *Proceedings of 13th Iranian Conference on Electrical Engineering*, Zanjan, Iran, May 2005.
- [23] M. S. Das, B. H. Bindhu, and A. Govardhan, “Evaluation of text detection and localization methods in natural images,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 6, pp. 277–282, 2012.
- [24] S. T. Deepa and S. P. Victor, “A novel method for text extraction,” *International Journal of Engineering Science Advanced Technology*, vol. 2, no. 4, pp. 961–964, 2013.
- [25] S. Li and J. T. Kwok, “Text extraction using edge detection and morphological dilation,” in *Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing (ISIMP ’04)*, pp. 330–333, IEEE, October 2004.
- [26] J. Poignant, L. Besacier, G. Quenot, and F. Thollard, “From text detection in videos to person identification,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME ’12)*, pp. 854–859, IEEE, July 2012.
- [27] R. Minetto, N. Thome, M. Cord, J. Fabrizio, and B. Marcotegui, “Snoopertext: a multiresolution system for text detection in complex visual scenes,” in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP ’10)*, pp. 3861–3864, IEEE, September 2010.
- [28] M. Anthimopoulos, B. Gatos, and I. Pratikakis, “Multiresolution text detection in video frames,” in *Proceedings of the 2nd International Conference on Computer Vision Theory and Applications (VISAPP ’07)*, pp. 161–166, March 2007.
- [29] C. Wolf and J.-M. Jolion, “Extraction and recognition of artificial text in multimedia documents,” *Pattern Analysis and Applications*, vol. 6, no. 4, pp. 309–326, 2004.
- [30] Y.-F. Pan, X. Hou, and C.-L. Liu, “A hybrid approach to detect and localize texts in natural scene images,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 800–813, 2011.
- [31] A. Gonzalez and L. M. Bergasa, “A text reading algorithm for natural images,” *Image and Vision Computing*, vol. 31, pp. 255–274, 2013.
- [32] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, “Scene text detection using graph model built upon maximally stable extremal regions,” *Pattern Recognition Letters*, vol. 34, pp. 107–116, 2012.
- [33] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, “Detecting texts of arbitrary orientations in natural images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’12)*, pp. 1083–1090, IEEE, June 2012.
- [34] O. J. Tobias and R. Seara, “Image segmentation by histogram thresholding using fuzzy sets,” *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1457–1465, 2002.
- [35] N. Senthilkumaran and R. Rajesh, “Edge detection techniques for image segmentation—a survey of soft computing approaches,” *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, pp. 250–254, 2009.
- [36] L. X. Wang, *A Course in Fuzzy Systems*, Prentice-Hall Press, Upper Saddle River, NJ, USA, 1999.
- [37] S. Tehsin, A. Masood, S. Kausar, and Y. Javed, “Text localization and detection method for born-digital images,” *IETE Journal of Research*, vol. 59, no. 4, pp. 343–349, 2013.
- [38] D. Karatzas, S. R. Mestre, J. Mas, F. Nourbakhsh, and P. P. Roy, “ICDAR 2011 robust reading competition—challenge 1: reading text in born-digital images (web and email),” in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR ’11)*, pp. 1485–1490, IEEE, September 2011.
- [39] C. Wolf and J.-M. Jolion, “Object count/area graphs for the evaluation of object detection and segmentation algorithms,” *International Journal on Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006.

Research Article

Hiding a Covert Digital Image by Assembling the RSA Encryption Method and the Binary Encoding Method

Kuang Tsan Lin¹ and Sheng Lih Yeh²

¹ Department of Mechanical and Computer Aided Engineering, St. John's University, 499 Sec. 4, Tam King Road, Tamsui, New Taipei City 25135, Taiwan

² Department of Mechanical Engineering, Lunghwa University of Science and Technology, 300 Sec. 1, Wanshou Road, Kueishan, Taoyuan County 33306, Taiwan

Correspondence should be addressed to Sheng Lih Yeh; slyeh@mail.lhu.edu.tw

Received 16 September 2013; Accepted 29 November 2013; Published 13 March 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 K. T. Lin and S. L. Yeh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Rivest-Shamir-Adleman (RSA) encryption method and the binary encoding method are assembled to form a hybrid hiding method to hide a covert digital image into a dot-matrix holographic image. First, the RSA encryption method is used to transform the covert image to form a RSA encryption data string. Then, all the elements of the RSA encryption data string are transferred into binary data. Finally, the binary data are encoded into the dot-matrix holographic image. The pixels of the dot-matrix holographic image contain seven groups of codes used for reconstructing the covert image. The seven groups of codes are identification codes, covert-image dimension codes, covert-image graylevel codes, pre-RSA bit number codes, RSA key codes, post-RSA bit number codes, and information codes. The reconstructed covert image derived from the dot-matrix holographic image and the original covert image are exactly the same.

1. Introduction

Image hiding methods can encode covert images in the space domain or in the spatial frequency domain. For space-domain encoding cases, hidden covert images can be reconstructed without any distortion usually, but their hiding security is lower often. On the other hand, for spatial-frequency-domain encoding cases, their hiding security is higher often, but there is more or less distortion for reconstructed covert images usually. Because spatial-frequency-domain encoding cases cannot reconstruct exact covert images, this paper will focus on space-domain encoding cases only. Of course, both higher security and higher noise-attack resistance are discussed here.

Many methods for hiding covert images in the space domain have been proposed. Image-transform method [1–4], cellular automata method [5], chaotic sequence method [6], image-scrambling method [7, 8], light separation method [9, 10], histogram shifting method [11, 12], and communication channel method [13] are some examples. All the methods

can work well, but most of them can be applied to printed or digital images only, whereas holographic images have become very popular and they appear on credit cards and bills. Therefore, other methods for hiding covert images on holographic holograms are needed. This paper will propose a hybrid hiding method to satisfy the requirement. The proposed method transforms a covert image with the Rivest-Shamir-Adleman (RSA) encryption method and encodes the transformed data into a dot-matrix holographic image with the binary encoding method.

The RSA encryption method is widely used in electronic document security. The RSA encryption method needs keys n , e , and d to encode and decode data [14–19]. The RSA encryption method usually uses very huge key n , which is, more than 100 bits, for assuring enough high security, so it is troublesome work to use the method. This paper will assemble the RSA encryption method and the binary encoding method [20] to allow keys with only tens of bits to be used and enough high security is still assured. The RSA encryption method includes three steps. Firstly, it

transforms a covert image to form a data string. Secondly, it transforms the data string to form a RSA encryption data string by using the standard RSA encryption method according to encryption keys. Finally, it transforms the RSA encryption data string to form an array. On the other hand, the binary encoding method uses binary data to denote all the element values of the RSA encryption string and encodes the binary data into a dot-matrix holographic image [21]. The pixels of the covert image contain seven groups of binary codes used for decoding the RSA encryption string. The seven groups of codes are identification codes, covert-image dimension codes, covert-image graylevel codes, pre-RSA bit number codes, RSA key codes, post-RSA bit number codes, and information codes. The RSA encryption string can be decoded directly from the dot-matrix holographic image.

Because the proposed hybrid hiding method possesses two layers of security (one layer is from the RSA encryption method and the other layer is from the binary encoding method), it can protect covert images very well. Unauthorized people can hardly reconstruct covert images correctly, but authorized people can easily reconstruct covert images correctly. Most of all, covert image reconstruction does not cause image distortion.

2. Theory

2.1. Reviewing the RSA Encryption Method. Assume that \mathbf{C} is an $M \times N$ 2^g -graylevel covert image for encoding, and assume that \mathbf{D} is an $M \times N$ matrix transformed by the RSA encryption method. For using the RSA encryption method, three keys (integers) are needed and the three keys are denoted by n , e , and d . The key n is an integer formed by the multiplication of two unequal prime numbers p and q ; that is,

$$n = p \times q. \quad (1)$$

Let an integer r be formed by

$$r = (p - 1) \times (q - 1). \quad (2)$$

Then the key e should be an integer between 2 and r , that is, $2 < e < r$, and the highest common factor of e and r is 1. The key d is derived when the remainder for the division $(e \times d)/r$ equals 1; that is,

$$1 \equiv (e \times d) \pmod{r}, \quad (3)$$

where the symbol “mod” denotes the modulus-after-division operation.

The keys for the encoding and decoding are used as below.

- (1) An integer s (before the RSA encoding) modulated by the RSA encryption must be smaller than n .
- (2) Another integer t (after the RSA encoding) is derived from the RSA encryption modulation of the integer s according to

$$t \equiv s^e \pmod{n}. \quad (4)$$

- (3) The integer s can be derived from the RSA encryption modulation of the integer t according to

$$s \equiv t^d \pmod{n}. \quad (5)$$

Equations (6a) and (6b) are an illustration to explain the RSA encryption processes for a 3×4 matrix. Equation (6a) is a matrix \mathbf{C} before the RSA encoding and (6b) is a matrix \mathbf{D} after the RSA encoding, where the keys n , e , and $d = 55, 13$, and 37). Every element s in \mathbf{C} and its corresponding element t in \mathbf{D} have the relationships as described by (4) and (5):

$$\mathbf{C} = \begin{bmatrix} 9 & 2 & 22 & 27 \\ 28 & 33 & 17 & 10 \\ 5 & 34 & 12 & 14 \end{bmatrix}, \quad (6a)$$

$$\mathbf{D} = \begin{bmatrix} 14 & 52 & 22 & 37 \\ 18 & 33 & 7 & 10 \\ 15 & 34 & 12 & 49 \end{bmatrix}. \quad (6b)$$

2.2. The Hybrid Hiding Method. Let \mathbf{T} be a $P \times Q$ matrix to encode an $M \times N$ covert image \mathbf{C} , and let \mathbf{H} be a $P \times Q$ holographic image for hiding the matrix \mathbf{T} . All the binary data encoded in the matrix \mathbf{T} are binary codes and they are categorized into seven groups of codes, identification codes, covert-image dimension codes, covert-image graylevel codes, pre-RSA bit number codes, RSA key codes, post-RSA bit codes, and information codes. Identification codes are used to judge if the codes encoded in the matrix \mathbf{T} belong to the binary encoding method or not; covert-image dimension codes M and N are used to denote the size of the covert image; covert-image graylevel codes 2^g are used to denote the graylevels of the covert image; pre-RSA bit number codes are used to denote the number of bits corresponding to every element of a data string before the RSA encryption; RSA key codes are used for the RSA encoding and decoding; and post-RSA bit number codes are used to denote the number of bits corresponding to every element of a data string after the RSA encryption. All the above six groups of codes are located at the first row of the matrix \mathbf{T} . Information codes are used to denote the binary data encoded from the data string after the RSA encryption and they are located at the second to final rows of the matrix \mathbf{T} .

To avoid using too many similar equations in this paper, an equation for plural uses is defined as

$$k = \sum_{i=1}^T k_i \cdot 2^{i-1} + k_0, \quad (7)$$

where k is an integer; all k_1, k_2, \dots , and k_T are 0 or 1; and k_0 is another integer.

Identification codes are a set of private binary codes used to judge if the data hidden in the covert image is encoded with the binary encoding method or not. The bit amount of the codes has to be big enough, for example, 1111001111001010110000110000 with 28 bits. The codes locate at bits 1~28 in the first row of the matrix \mathbf{T} .

Covert-image dimension codes M and N are used to derive the size $M \times N$ of the covert image, and they include

two sets of ten binary codes. The parameter M is indicated by the first set of ten binary codes M_1, M_2, \dots, M_9 , and M_{10} . The relationship of M and $M_1 \sim M_{10}$ is similar to (7), but every k has to be replaced by M , and $M_0 = 3$ and $T = 10$. The parameter N is indicated by the second set of ten binary codes N_1, N_2, \dots, N_9 , and N_{10} . The relationship of N and $N_1 \sim N_{10}$ is similar to (7), but every k has to be replaced by N , and $N_0 = 3$ and $T = 10$. The codes locate at bits 29~48 in the first row of the matrix \mathbf{T} .

Covert-image graylevel codes are used to derive the parameter g corresponding to the 2^g graylevels of the covert image. The parameter g is indicated by six binary codes g_1, g_2, g_3, g_4, g_5 , and g_6 . The relationship of g and $g_1 \sim g_6$ is similar to (7), but every k has to be replaced by g , and $g_0 = 1$ and $T = 6$. The codes locate at bits 49~54 in the first row of the matrix \mathbf{T} .

Pre-RSA bit number codes are used to denote the number of bits b corresponding to every element of a data string for the RSA encryption process. The parameter b is indicated by six binary codes b_1, b_2, b_3, b_4, b_5 , and b_6 . The relationship of b and $b_1 \sim b_6$ is similar to (7), but every k has to be replaced by b , and $b_0 = 1$ and $T = 6$. The codes locate at bits 55~60 in the first row of the matrix \mathbf{T} .

RSA key codes are used to denote the keys n and d . Although three keys n , e , and d are needed for encoding and decoding data strings, e is needed for the encoding only. The key n is indicated by thirty-five binary codes n_1, n_2, \dots, n_{34} , and n_{35} . The relationship of n and $n_1 \sim n_{35}$ is similar to (7), but every k has to be replaced by n , and $n_0 = 100$ and $T = 35$. The codes locate at bits 61~95 in the first row of the matrix \mathbf{T} . The key d is indicated by thirty-five binary codes d_1, d_2, \dots, d_{34} , and d_{35} . The relationship of d and $d_1 \sim d_{35}$ is similar to (7), but every k has to be replaced by d , and $d_0 = 2$ and $T = 35$. The codes locate at bits 96~130 in the first row of the matrix \mathbf{T} .

Post-RSA bit number codes are used to specify the number of bits a corresponding to every element of the RSA encryption data string. The parameter a is indicated by six binary codes a_1, a_2, a_3, a_4, a_5 , and a_6 . The relationship of a and $a_1 \sim a_6$ is similar to (7), but every k has to be replaced by a , and $a_0 = 2$ and $T = 6$. The codes locate at bits 131~136 in the first row of the matrix \mathbf{T} .

Information codes are used to reconstruct the RSA encryption data string for decoding the covert image \mathbf{C} .

2.3. Encoding Processes. The processes to encode the $M \times N$ covert image \mathbf{C} into the $P \times Q$ holographic image \mathbf{H} are explained below.

(1) Set a binary-data array \mathbf{R} with $1 \times Q$ elements. The elements of \mathbf{R} are copied from identification codes (for the first range), covert-image dimension codes (for the second range), covert-image graylevel codes (for the third range), pre-RSA bit number codes (for the fourth range), RSA key codes (for the fifth range), and post-RSA bit number codes (for the sixth range), whereas the other elements of \mathbf{R} not copied from the six groups of codes are all set as 0.

(2) The elements of the $M \times N$ covert image \mathbf{C} are copied to form the elements (from the left side to the right side) of a string \mathbf{A} with $M \times N$ elements from the first row to the last

row (the first priority) and from the left side to the right side (the second priority).

(3) Decompose every element $\mathbf{A}(k)$ of \mathbf{A} into g (because the covert image possesses 2^g graylevels) binary codes $c_i(k)$ ($i = 0, 1, \dots, g - 1$) according to

$$\mathbf{A}(k) = \sum_{i=0}^{g-1} c_i(k) \times 2^i. \quad (8)$$

(4) Create a binary-data string \mathbf{B} with $M \times N \times g$ elements according to

$$\mathbf{B}(i + g \times (k - 1)) = c_i(k), \quad (9)$$

where $1 \leq k \leq M \times N$ and $0 \leq i \leq g - 1$.

(5) Create a data string \mathbf{E} with $M \times N \times g/b$ elements from the $M \times N \times g$ elements of the binary-data string \mathbf{B} (all b bits of binary data are used to form an element of the data string for the RSA encryption) according to

$$\mathbf{E}(k) = \sum_{m'=1}^b \mathbf{B}(b \times (k - 1) + m') \times 2^{b-m'}, \quad (10)$$

where $1 \leq k \leq M \times N \times g/b$.

(6) Transform the data string \mathbf{E} with $M \times N \times g/b$ elements to a data string \mathbf{F} with $M \times N \times g/b$ elements by using (4) with the keys n and e .

(7) Decompose every element $\mathbf{F}(k)$ of \mathbf{F} into a binary codes $f_i(k)$ ($i = 0, 1, \dots, a - 1$) (all a bits of binary data are used to denote every element of the RSA encryption data string) according to

$$\mathbf{F}(k) = \sum_{i=0}^{a-1} f_i(k) \times 2^i, \quad (11)$$

where $1 \leq k \leq M \times N \times g \times a/b$.

(8) Create a binary-data string \mathbf{G} with $M \times N \times g \times a/b$ elements according to

$$\mathbf{G}(i + g \times (k - 1)) = f_i(k), \quad (12)$$

where $1 \leq k \leq M \times N \times g/b$ and $0 \leq i \leq a - 1$.

(9) Copy all of the $M \times N \times g \times a/b$ binary elements of \mathbf{G} (from the left side to the right side) to form a $(P-1) \times Q$ binary-data matrix \mathbf{S} (from the first row to the last row and from the left side to the right side). Since the amount $M \times N \times g \times a/b$ of the elements of \mathbf{G} is smaller than $(P-1) \times Q$, there are $(P-1) \times Q - M \times N \times g \times a/b$ elements in \mathbf{S} not copied from the elements of \mathbf{G} . The values of these extra elements are all set as 0.

(10) Combine the $1 \times Q$ binary-data array \mathbf{R} and the $(P-1) \times Q$ binary-data matrix \mathbf{S} to form a $P \times Q$ binary-data matrix \mathbf{T} . The first row of \mathbf{T} comes from \mathbf{R} , and the other rows of \mathbf{T} come from \mathbf{S} .

(11) Hide the $P \times Q$ binary-data matrix \mathbf{T} into the $P \times Q$ dot-matrix holographic image \mathbf{H} with the specified grating dot feature [21].

Equations (13a)–(13i) illustrate an example of the encoding processes. The binary-data array \mathbf{R} with 1×6 elements is

shown in (13a), the covert matrix \mathbf{C} is shown in (13b), the data string \mathbf{A} is shown in (13c), the binary-data string \mathbf{B} is shown in (13d), the data string \mathbf{E} is shown in (13e), the RSA encryption data string \mathbf{F} with the keys $(n, e) = (55, 13)$ is shown in (13f),

$$\mathbf{R} = [0 \ 1 \ 0 \ 1 \ 0 \ 1], \quad (13a)$$

$$\mathbf{C} = \begin{bmatrix} 2 & 7 \\ 3 & 6 \end{bmatrix}, \quad (13b)$$

$$\mathbf{A} = [2 \ 7 \ 3 \ 6], \quad (13c)$$

$$\mathbf{B} = [0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0], \quad (13d)$$

$$\mathbf{E} = [5 \ 13 \ 14], \quad (13e)$$

$$\mathbf{F} = [15 \ 8 \ 49], \quad (13f)$$

$$\mathbf{G} = [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1], \quad (13g)$$

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (13h)$$

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (13i)$$

2.4. Decoding Processes. The processes for reconstructing the covert image \mathbf{C}^* from the dot-matrix holographic image \mathbf{H} are explained below.

(1) Determine every element $\mathbf{T}(i, j)$ of the $P \times Q$ binary-data matrix \mathbf{T} from every grating dot feature of the $P \times Q$ dot-matrix holographic image \mathbf{H} according to the specified grating dot feature.

(2) Check the identification codes located at bits 1~28 in the first row of \mathbf{T} to judge whether the covert image \mathbf{H}^* contains codes encoded by the binary encoding method or not.

(3) According to the codes located at bits 29~136 in the first row of \mathbf{T} decode the parameters M, N, g, n, d, b , and a .

(4) Copy the first $M \times N \times g \times a/b$ elements in the second to final rows (from the top to the bottom and from the left side to the right side) of \mathbf{T} to form the $M \times N \times g \times a/b$ elements (from the left side to the right side) of the binary-data string \mathbf{G} .

(5) Create the data string \mathbf{F} with $M \times N \times g/b$ elements from the elements of the $M \times N \times g \times a/b$ binary-data matrix \mathbf{G} according to

$$\mathbf{F}(k) = \sum_{m'=1}^a \mathbf{G}(a \times (k-1) + m') \times 2^{a-m'}, \quad (14)$$

where $1 \leq k \leq M \times N \times g/b$.

the binary-data string \mathbf{G} is shown in (13g), the binary-data matrix \mathbf{S} is shown in (13h), and the binary-data matrix \mathbf{T} is shown in (13i):

(6) Transform the data string \mathbf{F} with $M \times N \times g/b$ elements to a data string \mathbf{E} with $M \times N \times g/b$ by using (5) with the keys n and d .

(7) Decompose every element $\mathbf{E}(k)$ of \mathbf{E} into b binary codes $h_i(k)$ ($i = 0, 1, \dots, a-1$) according to

$$\mathbf{E}(k) = \sum_{i=0}^{b-1} h_i(k) \times 2^i \quad (15)$$

and form a binary-data matrix \mathbf{B} with $M \times N \times g$ elements according to

$$\mathbf{B}(i(b-1) + k) = h_i(k), \quad (16)$$

where $0 \leq k \leq a-1$.

(8) Create a data string \mathbf{A} with $M \times N$ elements from the elements of the $M \times N \times g$ binary-data matrix \mathbf{B} according to

$$\mathbf{A}(k) = \sum_{m'=1}^g \mathbf{B}(g \times (k-1) + m') \times 2^{g-m'}, \quad (17)$$

where $1 \leq k \leq M \times N$.

(9) Create the $M \times N$ matrix \mathbf{C} from the data string \mathbf{A} with $M \times N$ elements according to

$$\mathbf{C}(i, j) = \mathbf{A}((i-1) \times N + j), \quad (18)$$

where $1 \leq i \leq M$ and $1 \leq j \leq N$.

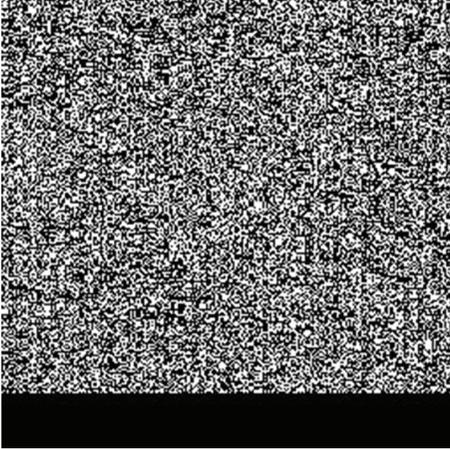


FIGURE 3: Binary-data matrix T for Figure 1(a).



FIGURE 4: Dot-matrix holographic image H for hiding the binary-data matrix T for Figure 1(a).

T for the covert image in Figure 1(a). The holographic image is on a Shipley 1813 photoresist plate and it contains 256×256 grating dots. All the grating dots have the same grating pitch of $1 \mu\text{m}$, but they have different grating orientations. The relation between the orientation $\theta(i, j)$ (measured from the x -axis) of a grating dot and the gray value $H(i, j)$ of a corresponding image pixel in Figure 2 is

$$\theta(i, j) = \frac{H(i, j) - 128}{1000} \times 180^\circ. \quad (19)$$

The grating dot pitch is about $63.5 \mu\text{m}$. The size of the holographic image is about $16.2 \text{ mm} \times 16.2 \text{ mm}$. The grating dots in the same column or row should be in a line for a dot-matrix hologram without hiding binary data, whereas the grating dots in the same column are in two separated lines for the dot-matrix hologram in Figure 4. The grating dots in the right line are used to denote the datum "1" and the grating dots in the left line are used to denote the datum "0". The distance between the two lines is about $15 \mu\text{m}$. All

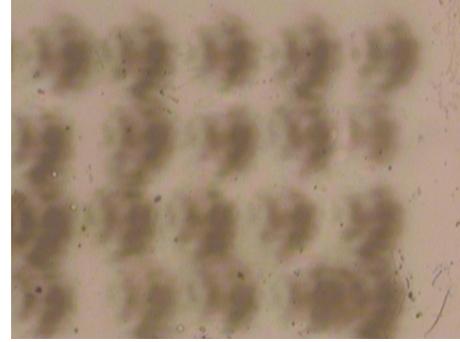


FIGURE 5: Some grating dots at the upper-right corner of the dot-matrix holographic image H for Figure 1(a).

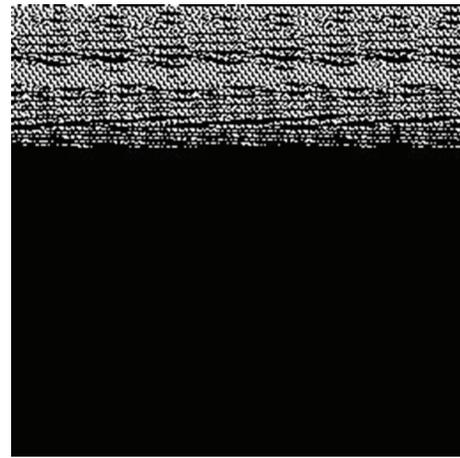


FIGURE 6: Binary-data matrix T for Figure 1(b).

the grating dots on the hologram in Figure 4 correctly hide the data in the binary-data matrix T for Figure 3. Figure 5 shows some of the grating dots at the upper-right corner of the dot-matrix holographic image. The original covert image C and the reconstructed covert image C^* derived from the holographic image H are exactly the same.

The second experiment for encoding Figure 1(b) is shown below. The first to 28th elements in \mathbf{R} are also specified as the identification codes, and they are set as 1111001111001010110000110000. The 29th to 48th elements in \mathbf{R} are specified as the covert-image dimension codes. Since the size of the covert image in Figure 1(b) is 128×128 , the dimension codes for M are 0001111101 (i.e., $128 = \sum_{i=1}^{10} M_i \cdot 2^{i-1} + 3$) and they are encoded at the 29th to 38th elements in \mathbf{R} ; the dimension codes for N are also 0001111101 and they are encoded at the 39th to 48th elements in \mathbf{R} . The 49th to 54th elements in \mathbf{R} are specified as the covert-image graylevel codes. Since the parameter g of the covert image used in this case is one (i.e., $2 = 2^1$), the codes are 000000 (i.e., $1 = \sum_{i=1}^6 g_i \cdot 2^{i-1} + 1$). The 55th to 60th elements (the fourth range) in \mathbf{R} are specified as the pre-RSA bit number codes. Because the selected pre-RSA bit number b in this case is eight, the 55th to 60th elements in \mathbf{R} are 000111 (i.e., $8 = \sum_{i=1}^6 b_i \cdot 2^{i-1} + 1$). The 61st to 130th elements in \mathbf{R}

encryption data string. The binary encoding method encodes the RSA encryption data string into binary codes and encodes the binary codes into the dot-matrix holographic image H . The reconstructed covert image C^* can be decoded directly from the dot-matrix holographic image H . Furthermore, there is no distortion for the decoding work of the covert image.

Conflict of Interests

The authors declare no conflict of interests regarding all the aspects related to this paper.

Acknowledgment

This paper was partially supported by the National Science Council of the Republic of China (Grant no. NSC 102-2221-E-129-004 and Grant no. NSC 101-2221-E-262-008).

References

- [1] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding: Techniques for Steganography and Digital Watermarking*, Artech House, Norwood, UK, 2000.
- [2] Y.-L. Yang, N. Cai, and G.-Q. Ni, "Digital image scrambling technology based on the symmetry of Arnold transform," *Journal of Beijing Institute of Technology*, vol. 15, no. 2, pp. 216–220, 2006.
- [3] Y. Zhou, S. Agaian, V. M. Joyner, and K. Panetta, "Two Fibonacci P-code based image scrambling algorithms," in *Image Processing: Algorithms and Systems VI*, vol. 6812 of *Proceedings of SPIE*, San Jose, Calif, USA, January 2008.
- [4] W. Chen and X. Chen, "Optical image encryption using multilevel Arnold transform and noninterferometric imaging," *Optical Engineering*, vol. 50, no. 11, Article ID 117001, 2011.
- [5] O. Lafe, "Data compression and encryption using cellular automata transforms," *Engineering Applications of Artificial Intelligence*, vol. 10, no. 6, pp. 581–591, 1997.
- [6] H. Zhang, J. Huang, and Z. Li, "New method of digital image scrambling based on binary tree generated by chaotic sequences," in *Remote Sensing and GIS Data Processing and Applications; and Innovative Multispectral Technology and Applications (MIPPR '07)*, vol. 6790 of *Proceedings of SPIE*, Wuhan, China, November 2007.
- [7] D. M. Wang, "The quasi-period of odd order magic square transformation on digital image," *Journal of Zhejiang University of Technology*, vol. 33, pp. 292–296, 2005.
- [8] K. T. Lin, "Information hiding based on binary encoding methods and pixel scrambling techniques," *Applied Optics*, vol. 49, no. 2, pp. 220–228, 2010.
- [9] S. K. Rajput and N. K. Nishchal, "Asymmetric color cryptosystem using polarization selective diffractive optical element and structured phase mask," *Applied Optics*, vol. 51, pp. 5377–5386, 2012.
- [10] W. Chen, X. Chen, A. Anand, and B. Javidi, "Optical encryption using multiple intensity samplings in the axial domain," *Journal of Optics Society of America A*, vol. 30, pp. 806–812, 2013.
- [11] L. C. Huang, L. Y. Tseng, and M. S. Hwang, "The study of data hiding in medical images," *International Journal of Network Security*, vol. 14, pp. 301–309, 2012.
- [12] L. C. Huang, L. Y. Tseng, and M. S. Hwang, "A reversible data hiding method by histogram shifting in high quality medical images," *Journal of Systems and Software*, vol. 86, no. 3, pp. 716–727, 2013.
- [13] Z. Liao, Y. Huang, and C. Li, "Research on data hiding capacity," *International Journal of Network Security*, vol. 5, pp. 140–144, 2007.
- [14] H. C. Williams, "A modification of the RSA public-key encryption procedure," *IEEE Transactions on Information Theory*, vol. 26, no. 6, pp. 726–729, 1980.
- [15] L. Kocarev, M. Sterjev, and P. Amato, "RSA encryption algorithm based on torus automorphisms," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. V-577–V-580, May 2004.
- [16] S. J. Aboud, M. A. Al-Fayoumi, M. Al-Fayoumi, and H. S. Jabbar, "An efficient RSA public key encryption scheme," in *Proceedings of the International Conference on Information Technology: New Generations (ITNG '08)*, pp. 127–130, April 2008.
- [17] R. S. Dhakar, A. K. Gupta, and P. Sharma, "Modified RSA Encryption Algorithm (MREA)," in *Proceedings of the 2nd International Conference on Advanced Computing and Communication Technologies (ACCT '12)*, pp. 426–429, January 2012.
- [18] N. Anane, M. Anane, H. Bessalah, M. Issad, and K. Messaoudi, "RSA based encryption decryption of medical images," in *Proceedings of the 7th International Multi-Conference on Systems, Signals and Devices (SSD '10)*, pp. 1–4, June 2010.
- [19] N. Saini and A. Sinha, "Key management of the double random-phase-encoding method using public-key encryption," *Optics and Lasers in Engineering*, vol. 48, no. 3, pp. 329–334, 2010.
- [20] K. T. Lin, "Digital information encrypted in an image using binary encoding," *Optics Communications*, vol. 281, no. 13, pp. 3447–3453, 2008.
- [21] S. L. Yeh, "Dot-matrix hologram with and encrypted figure," *Optical Engineering*, vol. 45, no. 9, Article ID 095801, 2006.
- [22] <http://www.ahead.com.tw>.

Research Article

The Nonlocal Sparse Reconstruction Algorithm by Similarity Measurement with Shearlet Feature Vector

Wu Qidi, Li Yibing, Lin Yun, and Yang Xiaodong

College of Information and Communication Engineering, Harbin Engineering University, Harbin, China

Correspondence should be addressed to Lin Yun; linyun@hrbeu.edu.cn

Received 24 August 2013; Revised 7 January 2014; Accepted 19 January 2014; Published 4 March 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Wu Qidi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the limited accuracy of conventional methods with image restoration, the paper supplied a nonlocal sparsity reconstruction algorithm with similarity measurement. To improve the performance of restoration results, we proposed two schemes to dictionary learning and sparse coding, respectively. In the part of the dictionary learning, we measured the similarity between patches from degraded image by constructing the Shearlet feature vector. Besides, we classified the patches into different classes with similarity and trained the cluster dictionary for each class, by cascading which we could gain the universal dictionary. In the part of sparse coding, we proposed a novel optimal objective function with the coding residual item, which can suppress the residual between the estimate coding and true sparse coding. Additionally, we show the derivation of self-adaptive regularization parameter in optimization under the Bayesian framework, which can make the performance better. It can be indicated from the experimental results that by taking full advantage of similar local geometric structure feature existing in the nonlocal patches and the coding residual suppression, the proposed method shows advantage both on visual perception and PSNR compared to the conventional methods.

1. Introduction

The reasonable representation is the basis of many tasks in image processing. The meaning of “reasonable” is to express the important information in image with less coefficients, which is called sparse representation [1]. We always want to express the signal in a cost-effective way, which could reduce the cost in signal processing. However, the emerging of sparse representation satisfied the requirements exactly. Wavelet is the landmark work, which is the most optimal expression of one-dimension signal. But it shows weak ability to express the high-dimension signal for its limited directions. For solving the problem, Donoho proposed the multiscale geometric analysis theory, which contains Ridgelet, Curvelet [2], Bandlet, and Contourlet [3] mainly, and then applied them to image restoration, which achieved good effect.

As for the sparse representation in image restoration, there are two main approaches, including multiscale geometric analysis (MGA) and dictionary learning. With the MGA, Donoho proposed the pioneer wavelet threshold shrinking method. After that, Yin designed a non-gaussian bivariate

Shrinkage function [4] under the MAP norm, which got the better performance. In 2006, as the limited directions in wavelet, Arthur proposed the NSCT transformation [5, 6] and applied it in the restoration area by Bayesian framework. But these conventional restoration models may not be accurate enough when the image is degraded seriously for their fixed over-completed dictionary, which is not sufficient to present the abundant structures in nature images. So, a new concept is proposed, which is called dictionary learning [7–9]. Besides, many methods about how to sparse coding are proposed, such as MP [10], OMP [11], and GP [12], which is called sparse coding. Further, a method called K-SVD [13, 14] is proposed by Aharon, which updates the dictionary by making SVD decomposition to the residual signal, and is applied to precision image restoration.

The goal of this paper is to research the sparse construction algorithm for image restoration based on the dictionary learning framework. We propose a nonlocal block sparse reconstruction scheme by shearlet feature vector, by which we measure the similarity between image patches and classify them for learning cluster dictionary. Then, we propose

a new objective function for sparse coding to realize the high accuracy image restoration.

2. Image Restoration with Dictionary Learning

Suppose that the model among the observation signal y , original signal x , and noise n satisfied

$$y = x + n. \quad (1)$$

According to the sparse representation theory in [1], the sparse coding is equal to solving the optimization problem in (2):

$$\begin{aligned} \min \quad & \|\alpha\|_0 \\ \text{s.t} \quad & \|y - D\alpha\|_2^2 \leq \varepsilon. \end{aligned} \quad (2)$$

In which, D is the sparse dictionary of x and α is the coding vector. Then, we can get the original signal by $\hat{x} = D\hat{\alpha}$. ε is a constant with the noise standard deviation, $\varepsilon = (C\sigma)^2$.

For the general restoration algorithm based on the sparse theory, dictionary D is fixed. So a new design for dictionary learning called K-SVD is proposed in reference [13] and good results have been achieved in image restoration. In the K-SVD, we implement the sparse coding and dictionary learning simultaneously. And, the optimization task in (2) can be changed to

$$(\hat{\alpha}_x, \hat{D}) = \arg \min_{\alpha, D} \{ \|y - D\alpha\|_2^2 + \lambda \|\alpha\|_0 \}, \quad \hat{x} = D\hat{\alpha}_x. \quad (3)$$

Elad et al. divided the image into many patches for not increasing the dictionary dimension and got the block-based sparse coding $\hat{\alpha}_{i,j}$ with

$$\hat{\alpha}_{i,j} = \arg \min \{ \|y_{i,j} - D\alpha\|_2^2 + \lambda \|\alpha\|_0 \}. \quad (4)$$

In which, $y_{i,j}$ presents the patch whose central is (i, j) . The size of patch is 7×7 . They are gained by sliding a squared windows in the image. Then, we can get the restored image by averaging these patches.

Generally, K-SVD learns the dictionary with some random training examples from the image, so the learned atoms show some weakness in presenting certain local structure in image, which makes the coding not sparsity enough. So, the paper has two contributions. The first is that we proposed a dictionary learning scheme based on clustering, which trained subdictionary for each cluster and then produced the universal dictionary with them. Secondly, we proposed a novel objective function for sparse coding, which added a coding residual item compared to the traditional ones. The detail for the two contributions will be introduced as follows.

3. A Novel Scheme for Image Restoration

3.1. Nonsampling Shearlet Transform. Nonsampling Shearlet [15] is a multiscale geometric analytical method advanced by Lim and Kutyniok. It consisted of laplacian

pyramid and direction filter bands with multidirections and shift invariant. For its parabola scale principle, Nonsampling Shearlet has excellent performances on capturing geometrical feature and singularity in high-dimension, which makes it widely used in image restoration.

3.2. Similarity Measurement. As to the many advantages with Shearlet, we propose a similarity measurement method. Implement the Shearlet transform with l levels and construct the feature vector for each pixel as

$$V = (sv^{1,0}, \dots, sv^{1,\varepsilon_1}, \dots, sv^{l,0}, \dots, sv^{l,\varepsilon_l})^T. \quad (5)$$

Here, ε_l is the number of subband in l th level and $sv^{l,\varepsilon}$ is the Shearlet coefficient. With the anisotropy feature, the coefficient for signal is large while the local geometric structure is similar to the basis function. Conversely, the coefficient is small, while the noise has the isotropy feature, so the coefficients are uniform in each primary function. Due to the above two reasons, the vector in (5) has the better antinoise ability, according to which, we take (5) to measure the similarity between the two patches. Suppose the size of patch is 5×5 , we get the vector in (5) for all the 25 pixels. Then, we construct some subvector $V_{\text{sub},k}$ generated by the k th dimension of V :

$$V_{\text{sub},k} = (V_1(k), V_2(k), \dots, V_{25}(k))^T. \quad (6)$$

In which, the $V_i(k)$ presents the k th element of V_i (i is the order number of the pixel in each patch, $i = 1, \dots, 25$). Suppose $V_{\text{sub},k}^1$ and $V_{\text{sub},k}^2$ are from two patches to be measured, we compute the index S_i as follows:

$$S_i = \exp \left\{ - \frac{\sum_k \|V_{\text{sub},k}^1 - V_{\text{sub},k}^2\|_2^2}{v_k^2} \right\}. \quad (7)$$

In which, v_k is called shearlet wave coefficient $v_k = \gamma\sigma_k$ (σ_k is the coefficient standard deviation with k th sub-band). With the formulation in (6), we can measure the similarity between two patches better than the Euclidean distance of gray in spatial domain.

3.3. Nonlocal Dictionary Learning

3.3.1. Nonlocal Dictionary Learning. The restoration algorithm consisted of two main parts that are dictionary learning and sparse coding, respectively. In this paper we improve the algorithm with the method for similarity measurement in Section 3.2. It is different to the conventional methods that we did not simply select the training patches from different example images to train the dictionary, while we select the nonlocal similar patches from the degraded image we want to restoration. And then, we cluster these patches to train the cluster subdictionary, which is more sufficient to make the universal dictionary better adapt to local geometrical feature than global K-SVD. Besides, the training data in the same cluster is similar to each other, so there is no need to produce an over-completed dictionary. So each subdictionary size is

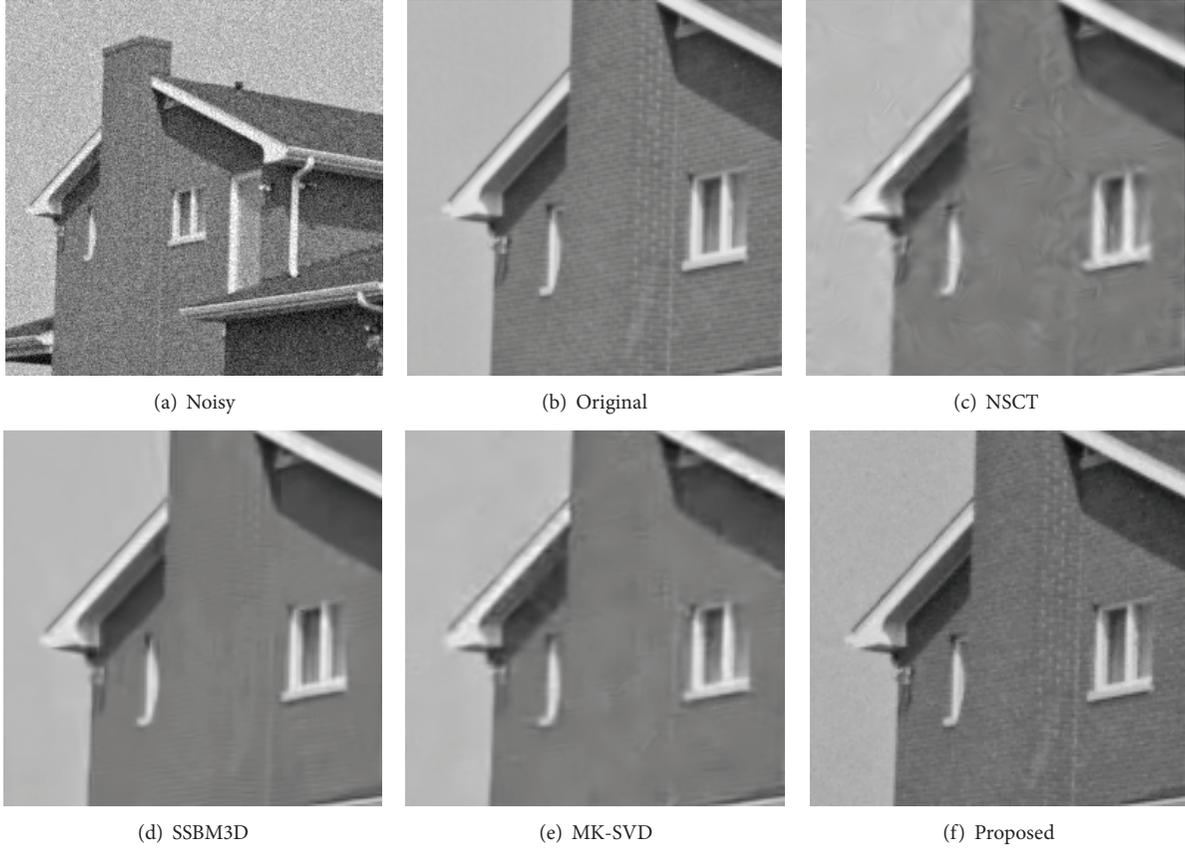


FIGURE 1: Restoration performance comparison on house ($\sigma = 15$).

half of the size of corresponding cluster. Now we present the concrete description for learning scheme as follows.

Step 1. Implement the Shearlet transform to the image Y .

Step 2. Construct the Shearlet feature vector according to the formula (5) (The patch size is 5×5).

Step 3. Calculate the standard deviation of k th subband ($\sigma_k = (1/N^2) \sum_{i=1}^{N^2} (y_i - \bar{y})^2$) and gain the wave coefficient v_k , y_i and \bar{y} is the shearlet coefficient and the mean value of coefficients; N is the total number of coefficients in k th subband.

Step 4. Cluster the patches by method in [16] with the index S_i in formula (6) and then produce the cluster Ω_i ($i = 1, 2, \dots, K$ is the cluster indicator).

Step 5. Take the patches in Ω_i as the training data and learning the cluster subdictionary with K-SVD.

Step 6. By cascading all the cluster dictionaries, we can get the universal dictionary for the whole image.

With the ability of capturing the local geometrical feature, the patches in the same set are highly similar by their geometry structure, which makes atoms in subdictionary have the strong adaptability to local structure. So, they can

sufficiently present the local geometrical feature. Cascaded by all the subdictionaries, the universal dictionary realizes the goal that presents all kinds of features in the whole image.

Now we show a simple example to show the advantage of clustering-based dictionary learning. We list a set of training examples:

$$\begin{aligned}
 & [100, 200, 0]^T, \\
 & [110 * \text{randn}(1), 120 * \text{randn}(1), 0]^T, \\
 & [100 * \text{randn}(1), 80 * \text{randn}(1), 0]^T, \\
 & [0, 150 * \text{randn}(1), 200 * \text{randn}(1)]^T.
 \end{aligned} \tag{8}$$

Here, “randn” means produce a Gaussian stochastic variable, and then, we take the above two groups of training data to learn a dictionary with two atoms. We adopt the proposed algorithm and K-SVD, respectively. Set the Maximum iteration number with KSVD to be 10 and we show the learned dictionary in Table 1.

As can be seen in Table 1, the atoms by proposed algorithm have the more similar structure with the training data.

3.3.2. Sparse Coding. For improving the coding accuracy, we proposed a novel optimization called coding residual

FIGURE 2: Restoration performance comparison on Barbara ($\sigma = 15$).

TABLE 1: Learned Dictionary with different methods.

Method	Learned Dictionary
KSVD	$\begin{bmatrix} -0.5630 & -0.0459 \\ -0.7995 & 0.2835 \\ -0.2096 & -0.9579 \end{bmatrix}$
	$\begin{bmatrix} -0.5375 & 0 \\ -0.8432 & 0.3294 \\ 0.0000 & -0.9442 \end{bmatrix}$
Proposed algorithm	

suppression optimization, which is more sufficient than l_1 -norm sparse coding. When dictionary is given, the l_1 -norm sparse coding for original signal can be realized as follows:

$$\hat{\alpha}_x = \arg \min \{ \|x - D\alpha\|^2 + \lambda \|\alpha\|_1 \}. \quad (9)$$

In the restoration problem, we can only get the observation y . So the sparse coding can be rewritten as

$$\hat{\alpha}_y = \arg \min \{ \|y - D\alpha\|^2 + \lambda \|\alpha\|_1 \}. \quad (10)$$

Comparing (9) and (10), we can see that if we want to reconstruct x with high accuracy, the coding $\hat{\alpha}_y$ is expected to be as close as possible to the true sparse coding $\hat{\alpha}_x$. So we introduce the residual coding α_δ as follows:

$$\alpha_\delta = \alpha_y - \alpha_x. \quad (11)$$

Then we change the object function (10) into the residual suppression form:

$$\hat{\alpha}_y = \arg \min \left\{ \|y - D\alpha\|^2 + \lambda \sum_k \|\alpha_k\|_1 + \mu \sum_k \|\alpha_\delta^k\|_1 \right\},$$

$$\alpha_\delta^k = \alpha_k - \hat{\eta}_i, \quad (12)$$

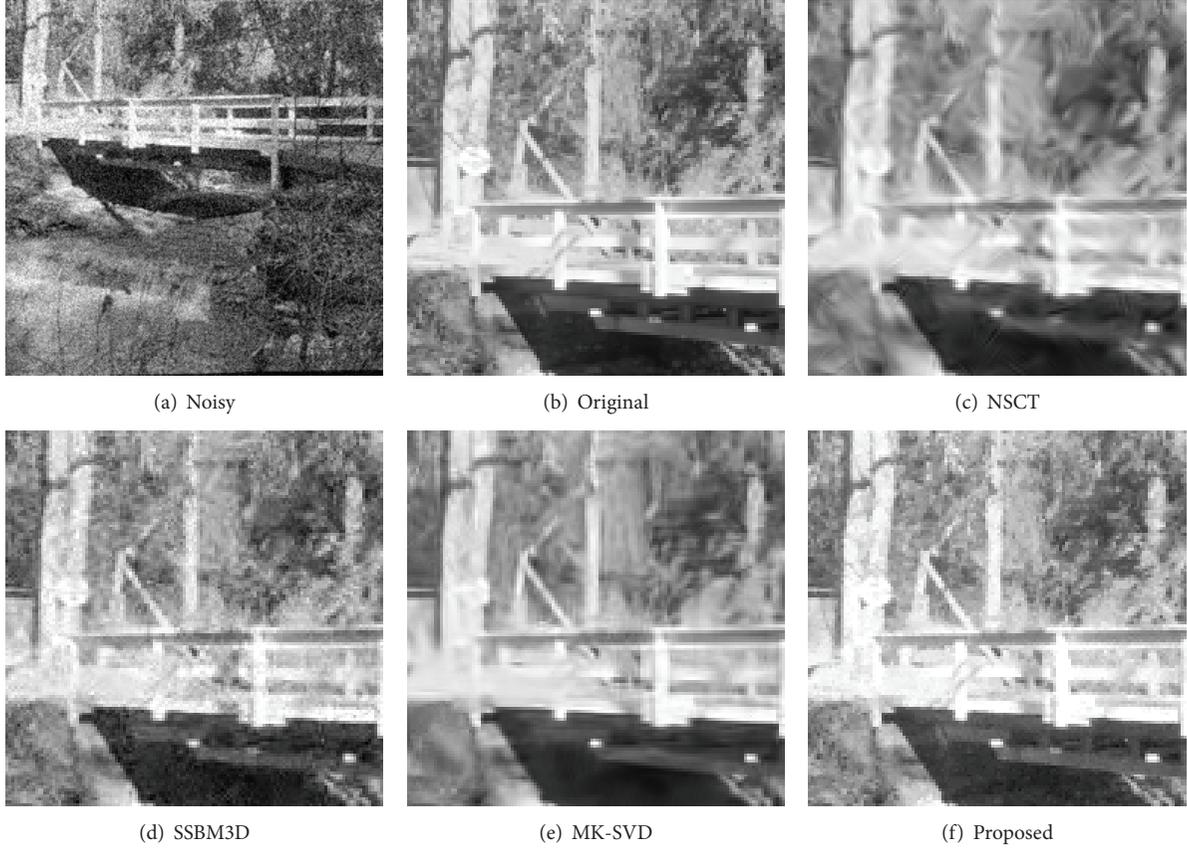
where k is the indicator of k th patch. In addition, we cannot get the true sparse coding practically, so the $\hat{\eta}_i$ is a good estimation of the true sparse coding and can be calculated from weighted average of sparse coding of patches in the same cluster; λ and μ are the regularization parameters. If k th patch is from Ω_i , we calculate $\hat{\eta}_i$:

$$\hat{\eta}_i = \sum_p \omega_{i,p} \hat{\alpha}_{i,p},$$

$$\omega_{i,p} = c \exp \left\{ -\|\hat{x}_i - \hat{x}_{i,p}\|_2^2 \right\} \quad (13)$$

(c is the normalization parameter).

The second term in optimization equation (12) is used to ensure the local sparsity that only part of atoms are selected for the dictionary. But, in our scheme, we select one subdictionary for each patch in a certain cluster, which

FIGURE 3: Restoration performance comparison on Bridge ($\sigma = 20$).

means that the coding of another subdictionary is zero. So our scheme guarantees the sparsity naturally; that is, we can move the second term in (12) and rewrite it as follows:

$$\hat{\alpha}_y = \arg \min \left\{ \|y - D\alpha\|^2 + \mu \sum_k \|\alpha_\delta^k\|_1 \right\}. \quad (14)$$

With (14), we can compact the optimization only with coding residual constraint, which means that the coding from the observed signal by (14) is close to that from the original signal by (9).

3.3.3. Scheme Summary. The whole scheme consisted of two-level iterative algorithm; now we present the brief steps as follows.

Initial: set the initial image $\hat{X} = Y$ (Y is the degraded image); $D^{(0)}$ is the DCT complete dictionary and calculate the initial sparse coding for each patch with any pursuit algorithm.

Outer Loop:

- (a) on the (OL)th iteration: learn the dictionary $D^{(OL)}$ by \hat{X} with the algorithm proposed in Section 3.3.1 (L is the maximum iterative number, $OL = 1, 2, \dots, L$);
- (b) compute the good estimation set $\{\eta_i\}_{i=1, \dots, K}$ for all the clusters under the $D^{(OL)}$.

(c) **Inner Loop:**

- (i) for each patch, we get its coding $\hat{\alpha}_y$ by (14), which can be solved with the method in [17];
- (ii) repeat (i) until all the patches are processed;

(d) estimate all the restoration patches by $\hat{x} = D^{(OL)}\hat{\alpha}_y$;

(e) $OL = OL + 1$, repeat.

On one hand, the $\hat{\eta}_i$ is more and more approach to the true sparse coding, which makes the residual suppression more sufficient. On the other hand, by alternating the sparse coding and dictionary learning, coding and dictionary are all improved and promote each other.

3.3.4. Parameter Selection. In conventional models, the regularization parameter is generally a constant. Hence, for making (14) more sufficient, we give the derivation of self-adaptive regularization parameter μ under the Bayes framework.

For written convenience, we take $\delta = \alpha_k - \hat{\eta}_i$ as the residual coding vector. Under the Bayes framework, the MAP estimation of δ can be computed as

$$\begin{aligned} \delta &= \arg \max \{ \log(p(\delta | y)) \} \\ &= \arg \max \{ \log(p(y | \delta)) + \log(p(\delta)) \}. \end{aligned} \quad (15)$$

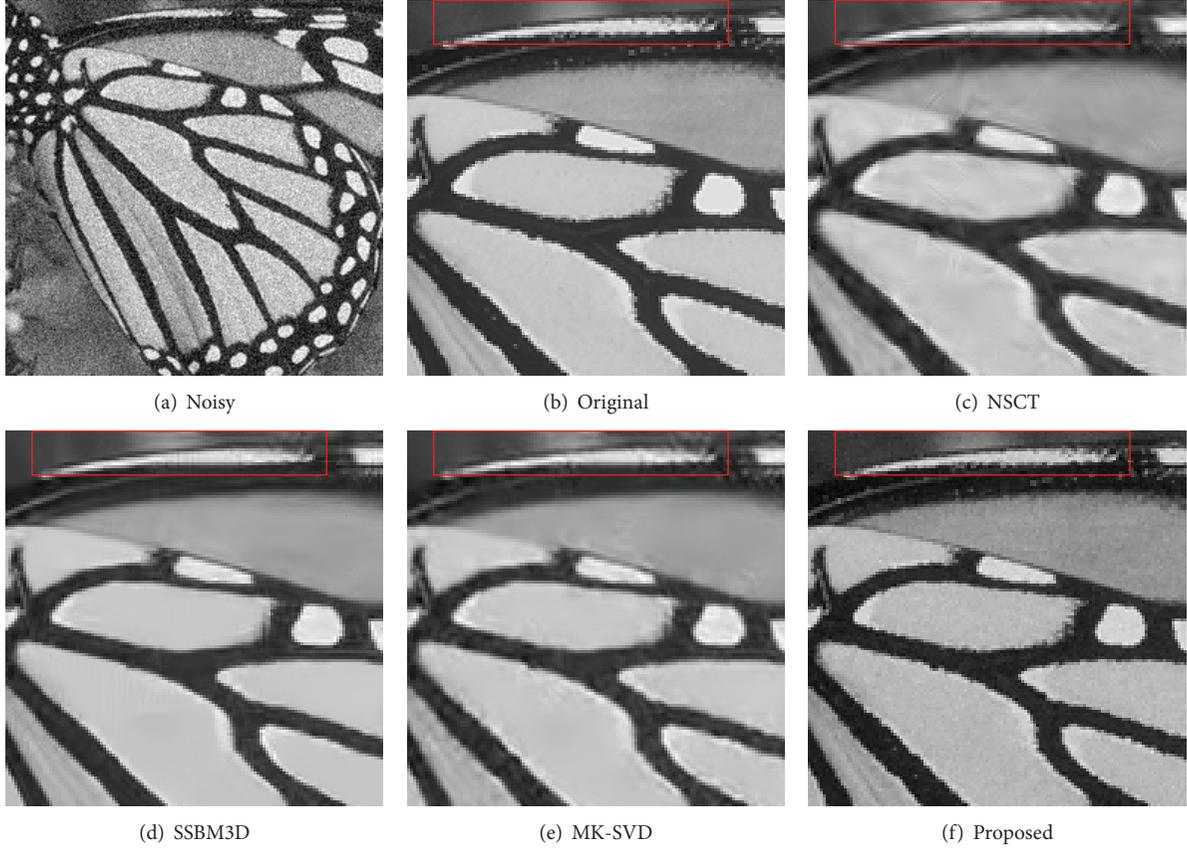


FIGURE 4: Restoration performance comparison on Butterfly ($\sigma = 20$).

And the $p(y | \delta)$ is

$$p(y | \delta) = p(y | \alpha, \hat{\eta}) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp \left\{ -\frac{1}{2\sigma_n^2} \|y - D\alpha\|_2^2 \right\}, \quad (16)$$

where σ_n is the standard deviation of noise.

As for $p(\delta)$, with some statistics experiments, we gain the experience model with the i.i.d Laplacian distribution:

$$\begin{aligned} p(\delta) &= \prod_k \prod_i p(\delta_k(i)) \\ &= \prod_k \prod_i \frac{1}{2\sigma_{k,i}} \exp \left\{ -\frac{1}{\sigma_{k,i}} |\delta_k(i)| \right\}. \end{aligned} \quad (17)$$

In which, $\delta_k(i)$ is the i th element of the residual vector of the k th patch and $\sigma_{k,i}$ is the standard deviation of $\delta_k(i)$.

Combining (15), we can obtain the following:

$$\hat{\alpha}_y = \arg \min \left\{ \frac{1}{2\sigma_n^2} \|y - D\alpha\|_2^2 + \frac{1}{\sigma_{k,i}} \sum_k \sum_i |\delta_k(i)| \right\}. \quad (18)$$

For a given $\hat{\eta}_i$ ($\alpha_k = \hat{\eta}_i + \delta_k$), the optimal sparse coding α can be obtained as follows:

$$\hat{\alpha}_y = \arg \min \left\{ \|y - D\alpha\|_2^2 + \frac{2\sigma_n^2}{\sigma_{k,i}} \sum_k \sum_i |\delta_k(i)| \right\}. \quad (19)$$

Compared to the regularization in (14), we can set the μ to be the following self-adaptive form:

$$\mu_{k,i} = \frac{2\sigma_n^2}{\sigma_{k,i}}. \quad (20)$$

4. Experimental Results and Analysis

To verify the performance of the algorithm proposed in the paper, we show some contrast restoration experiments for image denoising. The contrast algorithms are, respectively, NSCT method in [6], SSBM3D in [18], MK-SVD in [14], and the proposed algorithm in this paper. The noisy image is generated by adding the Gaussian noise with different standard deviation ($\sigma = 15, 20$). The size of test images is all 256×256 and we set the parameters as follows.

$\varepsilon = 3$, $l = 3$ (mentioned in Section 3.2); $K = 50$ (mentioned in Section 3.3.1); $p = 5$, (mentioned in Section 3.3.2).

To show the objective evaluation for restoration image, we take the PSNR as the indicator for different algorithms. Meanwhile, we show part of the experiment results to view the denoising performance. Suppose x and \hat{x} are the original

TABLE 2: The PSNR (dB) indicator with $\sigma = 15$.

	Noisy	NSCT	BM3D	K-SVD	Proposed
House	24.89	33.11	34.67	33.92	34.22
Barbara	24.63	30.49	32.33	31.40	31.97
Peppers	24.61	31.51	32.70	31.77	32.16
Bridge	24.70	27.71	28.83	28.55	28.62
Butterfly	24.63	29.17	30.64	29.89	30.18

TABLE 3: The PSNR (dB) indicator with $\sigma = 20$.

	Noisy	NSCT	BM3D	K-SVD	Proposed
House	22.10	31.78	33.76	32.67	32.89
Barbara	22.05	28.95	30.68	29.75	30.19
Peppers	22.09	30.12	31.36	30.46	31.12
Bridge	22.13	26.41	27.25	26.87	27.02
Butterfly	21.11	27.65	29.46	28.32	28.96

image and restoration image, respectively, the definition of PSNR is

$$\text{PSNR} = 10 \log \frac{255^2}{\|x - \hat{x}\|_2^2}. \quad (21)$$

Figures 1 and 2 are the restoration performance with the standard deviation $\sigma = 15$, while Figures 3 and 4 are the performance with $\sigma = 20$. As for the other images we only show the PSNR value reported in Tables 2 and 3.

For each group, by the local detail image, we can get some conclusion. Though all the four methods can achieve the denoising task, there are some differences between each other. For the NSCT method, the ability of capturing structure is not adaptive to different image due to its fixed dictionary. Additionally, some scratches appeared in the restoration results, which did not exist in the original image and the PSNR is lowest among the four methods. Compared to the other method, BM3D shows advantage on PSNR value. But its restoration results are smoothed excessively, which leads to lose much detail texture. Owing to the dictionary that is learned by the random example patches from different images, the K-SVD algorithm generates some visual artifacts and cannot recover the local geometry feature sufficiently. So the restoration results of K-SVD are only better than the NSCT in the four methods. The proposed method in this paper shows advantage both on the restoration performance and PSNR value. Though its PSNR is lower than the BM3D, the detail recovered ability is stronger than the other three methods.

5. Conclusion

Owing to that the abundant geometry information and the self-similarity are the important influencers to be utilized in the image restoration by sparse representation, the paper proposed a novel scheme that restores the image by the nonlocal sparse reconstruction with similarity measurement. We cluster the patches from the degraded image by similarity measurement with Shearlet feature vector, which is good at

capturing the local geometry structure in image and then take them to train the cluster dictionary and get good estimation of true sparse coding for original image. Additionally, we also show the derivation of regularization parameter under the Bayes framework. By the cluster dictionary learning and the coding residual suppression, the proposed scheme shows advantages both on the visual performance and PSNR value compared to the leading denoising methods. Besides the image denoising, the proposed model can be extensive to other restoration tasks such as deblurring and superresolution.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Nation Nature Science Foundation of China nos. 61201237 and 61301095, Nature Science Foundation of Heilongjiang Province of China no. QC2012C069, and the Fundamental Research Funds for the Central Universities no. HEUCFZ1129. Meantime, this paper is funded by the Defense prereseach foundation Project of shipbuilding industry science no. 10J3.1.6.

References

- [1] M. Alferdo, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *Society for Industrial and Applied Mathematics*, vol. 51, no. 1, pp. 34–81, 2009.
- [2] G. G. Bhutata, R. S. Anand, and S. C. Saxena, "Edge preserved image enhancement using adaptive fusion of images denoised by wavelet and curvelet transform," *Digital Signal Processing*, vol. 21, no. 1, pp. 118–130, 2011.
- [3] L. Shang, P.-G. Su, and T. Liu, "Denoising MMW image using the combination method of contourlet and KSC shrinkage," *Neurocomputing*, vol. 83, no. 15, pp. 229–233, 2012.

- [4] S. Yin, L. Cao, Y. Ling, and G. Jin, "Image denoising with anisotropic bivariate shrinkage," *Signal Processing*, vol. 91, no. 8, pp. 2078–2090, 2011.
- [5] A. L. da Cunha, J. Zhou, and M. N. Do, "The nonsubsampling contourlet transform: theory, design, and applications," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3089–3101, 2006.
- [6] J. Jia and L. Chen, "Using normal inverse Gaussian model for image denoising in NSCT domain," *Acta Electronica Sinica*, vol. 39, no. 7, pp. 1563–1568, 2011.
- [7] R. Rubinfeld, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.
- [8] Y. He, T. Gan, W. Chen, and H. Wang, "Multi-stage image denoising based on correlation coefficient matching and sparse dictionary pruning," *Signal Processing*, vol. 92, no. 1, pp. 139–149, 2012.
- [9] K. Labusch, E. Barth, and T. Martinetz, "Soft-competitive learning of sparse codes and its application to image reconstruction," *Neurocomputing*, vol. 74, no. 9, pp. 1418–1428, 2011.
- [10] Z. Xue, D. Han, and J. Tian, "Fast and robust reconstruction approach for sparse fluorescence tomography based on adaptive matching pursuit," in *Proceedings of the IEEE Conference on Asia Communications and Photonics*, pp. 1–6, November 2011.
- [11] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE Journal on Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 310–316, 2010.
- [12] T. Blumensath and M. E. Davies, "Stagewise weak gradient pursuits," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4333–4346, 2009.
- [13] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [14] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *Multiscale Modeling and Simulation*, vol. 7, no. 1, pp. 214–241, 2008.
- [15] W.-Q. Lim, "The discrete shearlet transform: a new directional transform and compactly supported shearlet frames," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1166–1180, 2010.
- [16] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
- [17] I. Daubechies, M. Defrise, and C. de Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.
- [18] M. Poderico, S. Parrilli, G. Poggi, and L. Verdoliva, "Sigmoid shrinkage for BM3D denoising algorithm," in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP '10)*, pp. 423–426, October 2010.

Research Article

Robust Homography Estimation Based on Nonlinear Least Squares Optimization

Wei Mou,¹ Han Wang,¹ and Gerald Seet²

¹ School of Electrical & Electronics Engineering, Nanyang Technological University, Singapore

² School of Mechanical & Aerospace Engineering, Nanyang Technological University, Singapore

Correspondence should be addressed to Wei Mou; mouwei@ntu.edu.sg

Received 24 October 2013; Accepted 14 January 2014; Published 26 February 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Wei Mou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The homography between image pairs is normally estimated by minimizing a suitable cost function given 2D keypoint correspondences. The correspondences are typically established using descriptor distance of keypoints. However, the correspondences are often incorrect due to ambiguous descriptors which can introduce errors into following homography computing step. There have been numerous attempts to filter out these erroneous correspondences, but it is unlikely to always achieve perfect matching. To deal with this problem, we propose a nonlinear least squares optimization approach to compute homography such that false matches have no or little effect on computed homography. Unlike normal homography computation algorithms, our method formulates not only the keypoints' geometric relationship but also their descriptor similarity into cost function. Moreover, the cost function is parametrized in such a way that incorrect correspondences can be simultaneously identified while the homography is computed. Experiments show that the proposed approach can perform well even with the presence of a large number of outliers.

1. Introduction

Estimating homography given keypoint correspondences of image pairs has received much attention due to its extensive applications, for example, panoramas generation [1], motion estimation [2], camera calibration [3], and augmented reality [4].

The homography estimation given an image pair can be decomposed into two stages. In the first stage, keypoints in two images are detected and their local image descriptors are extracted. The keypoint matches are established by comparing the corresponding descriptors. After that, the false matches, due to ambiguous descriptors, are detected and removed using robust methods. In the second stage, a cost function based on remaining matches is defined and the homography is computed by minimizing the cost function.

Much attention has been paid to the first stage. The descriptor ambiguity and reliability can be improved by using more distinctive descriptors such as PCA-SIFT [5]. However, it cannot completely resolve ambiguities especially in the case of repetitive patterns or occlusions.

Hypothesize-and-verification framework such as random sample consensus (RANSAC) [6] is the most popular method to remove inevitable outliers. It randomly and repeatedly selects and verifies a small set of matches to find inliers that are consistent with some global geometric constraints.

More sophisticated variations have been proposed to improve the standard RANSAC algorithm. Some approaches such as PROSAC [7] and ARRISAC [8] focus on improving the reliability of hypothesis. To achieve this, these methods use image appearance to select keypoint correspondences with high confidence and, consequentially, speed up the search for consistent matches. Some other approaches such as MSAC [9], MLESAC [9], and MAPSAC [10] use more sophisticated measures such as likelihood or posterior instead of point consensus to better verify hypothesis. These approaches have succeeded in greatly reducing the error rate.

In order to estimate homography that can best describe the data, usually all inliers are then fed into the optimization process. Given point correspondences, a cost function is formulated based on the difference between the measured and estimated image coordinates. Homography is obtained

by minimizing this cost function using Direct Linear Transformation (DLT) or by iterative optimization methods such as Gauss-Newton or Levenberg-Marquardt. However, after the initial selection stage, it cannot be guaranteed that the data is noise-free before minimizing the cost function as there is no robust rule to define outliers. Conventional least squares optimization algorithms are in general not robust to outliers. The reason is that Gaussian distribution of noise is the basic assumption for solving least squares problem and Gaussian distribution is sensitive to outliers due to its narrow tailed nature. Error increases quadratically in least squares problem. Hence, the incorrect correspondences can easily lead to divergence of homography estimation. For example, a false point matching causes big geometric difference on the measured and estimated image coordinates. In order to compensate for this error, the homography has to deviate from its true value during the optimization process. Moreover, in some applications such as with ego motion estimation or circular panorama generation, such errors accumulate over time. Hence, even an extremely low error rate will eventually result in significant errors. As a result, given point correspondences, a robust cost function that can accurately calculate homography in the presence of outliers is needed. A desirable property for such an algorithm is that the incorrect correspondences have no or little contribution to the optimization process. In other words, during the homography calculation process, the algorithm should be able to identify the undetected mismatches as well.

Some robust cost functions such as Huber [11] have been proposed to reduce errors introduced by outliers. Unlike normal least square in which the error has a quadratic influence on the cost function, the Huber function makes cost increase linearly if the error exceeds a certain threshold. This means that it weighs large error less. However, the Huber cost function is not enough to deal with outliers, because the influence of outliers is reduced rather than removed.

Serradell et al.'s method [12] is able to solve correspondences and homography simultaneously. They combine geometric prior and appearance prior to achieve this. More specifically, the search space of the homography is constrained such as to limit the range of rotations and scales. Several homography hypotheses are sampled in this search space. Gaussian Mixture Models that best fit these samples are formed as the geometric prior. Appearance prior is based on descriptor similarity distances. The homography estimate and its covariance are iteratively updated by a Kalman filter that uses the best correspondences as measurements until the covariance becomes negligible. The potential mismatches are detected as those that are least likely to reduce the covariances of the Kalman filter. In such a way, the influence of mismatches is removed instead of reduced which makes it robust to high numbers of outliers. Their method can also be categorized into hypothesis-and-verify approach.

In this paper, we propose a robust homography estimation approach that is different from the hypothesis-and-verify approach. It allows the homography calculation and incorrect correspondences identification simultaneously within a Bayesian framework. The contributions of this paper are as follows.

- (i) No initial samples or hypothesis of Homography are required.
- (ii) We formulate a new cost function which integrates keypoint consensus and descriptor similarity.
- (iii) We introduce a new set of parameters that represent the confidence for each correspondence, by solving which using nonlinear optimization, both of the homography and false keypoint correspondences can be determined simultaneously.

With these improvements, our method can achieve robustness even when a rather high percentage of correspondences are incorrect. The influence of outliers for homography computing is much reduced, even removed, and thus can output a more accurate homography that satisfies most pixels' geometric relationship between two images. Experimental results show the effectiveness and efficiency of the proposed approach in both synthetic images and real-life images.

2. Cost Function Formulation

We first give some notations and briefly review the conventional formulation of homography estimation before introducing our extension.

Given an image pair, \mathbf{x}_i^j represents the i th point in j th image and its inhomogeneous image coordinate is denoted as (x_i^j, y_i^j) . The data set of extracted keypoint correspondences is \mathbf{M} . Each pair of corresponding points \mathbf{x}_k^1 and \mathbf{x}_l^2 defines a single point \mathbf{m}_i in a measurement space \mathfrak{R}^4 , formed by joining the coordinates in each image. The i th keypoint correspondence of image pair is represented by vector $\mathbf{m}_i = (x_k^1, y_k^1, x_l^2, y_l^2)^T$. The homography that transforms the first image to the second image is \mathbf{H} . According to Bayesian framework, \mathbf{H} , is estimated as

$$\Pr(\mathbf{H} | \mathbf{M}) = \frac{\Pr(\mathbf{M} | \mathbf{H}) \Pr(\mathbf{H})}{\Pr(\mathbf{M})}. \quad (1)$$

There is no prior knowledge about matchings and homography, we assume them to be uniformly distributed which makes $\Pr(\mathbf{H})$ and $\Pr(\mathbf{M})$ trivial to the optimization problem. Also, because feature matching pairs are determined using descriptor similarity only, we assume them to be conditionally independent from each other. Hence, (1) is changed into

$$\Pr(\mathbf{H} | \mathbf{M}) \propto \Pr(\mathbf{M} | \mathbf{H}) \propto \prod_{i=1 \dots n} \Pr(\mathbf{m}_i | \mathbf{H}), \quad (2)$$

where n is the number of correspondences. Let $f(\mathbf{x}_k^1, \mathbf{H})$ be the estimation of point \mathbf{x}_k^1 projected on image 2 from image 1 by \mathbf{H} . The geometric error vector between the estimated and measured image coordinates introduced by matching \mathbf{m}_i is defined as

$$\mathbf{e}_i = \mathbf{x}_l^2 - f(\mathbf{x}_k^1, \mathbf{H}). \quad (3)$$

Without loss of generality, the noise in the two images is assumed to be Gaussian on each image coordinate with zero

mean and covariance matrix Σ . Equation (2) can be written as

$$\begin{aligned} \Pr(\mathbf{M} | \mathbf{H}) &= \prod_{i=1:n} \left(\frac{1}{\sqrt{2\pi} |\Sigma|} \right) \exp\left(-\frac{1}{2} \mathbf{e}_i^T \Sigma^{-1} \mathbf{e}_i\right) \\ &= \left(\frac{1}{\sqrt{2\pi} |\Sigma|} \right)^n \prod_{i=1:n} \exp\left(-\frac{1}{2} \mathbf{e}_i^T \Sigma^{-1} \mathbf{e}_i\right), \end{aligned} \quad (4)$$

where n is the number of correspondences and $|\Sigma|$ is the determinant of covariance. Take negative log likelihood of all the correspondences; (4) becomes

$$-\log \Pr(\mathbf{M} | \mathbf{H}) = \eta + \frac{1}{2} \sum_{i=1:n} \mathbf{e}_i^T \Sigma^{-1} \mathbf{e}_i, \quad (5)$$

where η is a constant number. In order to optimally estimate \mathbf{H} , from (1) to (5), a MAP estimate \mathbf{H}^* is made such that

$$\begin{aligned} \mathbf{H}^* &= \arg \max_{\mathbf{H}} \Pr(\mathbf{H} | \mathbf{M}) \\ &= \arg \min_{\mathbf{H}} -\log \Pr(\mathbf{M} | \mathbf{H}) \\ &= \arg \min_{\mathbf{H}} \underbrace{\sum_{i=1:n} \mathbf{e}_i^T \Sigma^{-1} \mathbf{e}_i}_{\text{cost function } F(x)}, \end{aligned} \quad (6)$$

where minimizing cost function $F(x)$ is a nonlinear least squares problem and it can be solved using iterative methods such as Gauss-Newton or Levenberg-Marquardt. However, in this cost function, there is no mechanism to deal with outliers. Thus, false correspondences can easily cause the least squares optimization to converge to a wrong estimation.

It is desirable to reduce or ignore the influence of incorrect correspondences during optimization. Our main approach to achieve this is the following: as the optimization proceeds, we dynamically assign a corresponding reliability to each point correspondence. In other words, the contribution of each correspondence to the optimization problem changes while the optimization proceeds. This leads to an extended optimization problem: not only parameters of homography need to be optimized but also the reliability of feature correspondence. Referring to cost function, Σ in (6) represents the uncertainty of the measured image coordinates which is a good measure to control the portion of contribution of the corresponding point correspondence. For example, high uncertainty means the noise in the measured image coordinate is high and thus the error introduced by the corresponding keypoint match will contribute little to the optimization problem. However, in (6) we can observe that Σ is constant for all correspondences. In order to make it change dynamically, we introduce a new set of parameters \mathbf{C} to be optimized which describes the covariance matrix Σ_i for each measurement.

Moreover, the similarity of corresponding keypoints from two views is also included into cost function. The collection of similarity score is denoted as \mathbf{S} and \mathbf{s}_i is the similarity score of correspondence \mathbf{m}_i . The descriptor similarity is integrated into optimization problem in such a way that it helps to

determine the values in \mathbf{C} which describe the confidences of correspondences and \mathbf{c}_i represents the confidence for match \mathbf{m}_i . With the idea of our approach being introduced, (1) can be extended as

$$\begin{aligned} \Pr(\mathbf{H} | \mathbf{M}, \mathbf{C}, \mathbf{S}) &\propto \Pr(\mathbf{M}, \mathbf{C} | \mathbf{H}, \mathbf{S}) \\ &\propto \Pr(\mathbf{M} | \mathbf{C}, \mathbf{H}) \Pr(\mathbf{C} | \mathbf{S}) \\ &\propto \prod_{i=1:n} \frac{\Pr(\mathbf{m}_i | \mathbf{c}_i, \mathbf{H})}{\text{likelihood}} \frac{\Pr(\mathbf{c}_i | \mathbf{s}_i)}{\text{prior}}. \end{aligned} \quad (7)$$

The first term is the likelihood of matching given the homography and uncertainty of measurements, while the second term is the prior probability of \mathbf{c}_i and the uncertainty of which is based on \mathbf{s}_i .

With Gaussian assumption, new cost function to be minimized can be obtained by taking the negative log likelihood of (7):

$$F'(x) = \sum_{i=1:n} \left(\frac{\mathbf{e}_i^T \Sigma_i^{-1} \mathbf{e}_i}{\delta_i} + \frac{\boldsymbol{\varepsilon}_i^T \Psi_i^{-1} \boldsymbol{\varepsilon}_i}{\rho_i} \right), \quad (8)$$

with Ψ_i corresponding to covariance of prior distribution of \mathbf{c}_i . $\boldsymbol{\varepsilon}_i$ is the difference between the current \mathbf{c}_i and its expected value $\bar{\mathbf{c}}_i$. We set $\bar{\mathbf{c}}_i$ to 1 for each correspondence.

Next, we elaborate on how does \mathbf{c}_i describe Σ_i . To simplify the problem, we assume that the error distribution of measurements along x and y directions are independent from each other and their standard deviations are the same. Thus, \mathbf{c}_i is defined as a scalar and it is enough to describe Σ_i . As a result, δ_i in (8) becomes

$$\delta_i = \mathbf{e}_i^T \begin{bmatrix} \mathbf{c}_i^{-2} & 0 \\ 0 & \mathbf{c}_i^{-2} \end{bmatrix}^{-1} \mathbf{e}_i = (\mathbf{c}_i \cdot \mathbf{e}_i)^T \mathbf{I}_{2 \times 2} (\mathbf{c}_i \cdot \mathbf{e}_i), \quad (9)$$

where \mathbf{c}_i is defined as the inverse of the standard deviation of Σ_i . However, in (9) we can see that it can also be interpreted as the weight of \mathbf{e}_i which is the geometric error introduced by correspondence \mathbf{m}_i . If \mathbf{m}_i is an outlier, the corresponding \mathbf{e}_i will contribute little to optimization problem by assigning a small weight to it and will be ignored if $\mathbf{c}_i = 0$, because in this case, \mathbf{e}_i will not be added to the global error term. In this interpretation, if we scale \mathbf{c}_i to $[0, 1]$, \mathbf{c}_i not only makes optimization process robust to outliers but also works as outlier and inlier selector as false \mathbf{m}_i will make \mathbf{c}_i close to 0, while correct \mathbf{m}_i makes it close to 1.

Similarly as δ_i , we can write ρ_i in (8) as

$$\rho_i = (\mathbf{s}_i \cdot \boldsymbol{\varepsilon}_i)^T \mathbf{I}_{2 \times 2} (\mathbf{s}_i \cdot \boldsymbol{\varepsilon}_i). \quad (10)$$

The local feature descriptor pair of detected keypoints in \mathbf{m}_i is $(\boldsymbol{\varphi}_k^1, \boldsymbol{\varphi}_l^2)$ in which $\boldsymbol{\varphi}_k^j$ is the descriptor vector of k th keypoint in image j . Let $\text{dist}(\boldsymbol{\varphi}_k^1, \boldsymbol{\varphi}_l^2)$ represent the Euclidean distance between descriptors $\boldsymbol{\varphi}_k^1$ and $\boldsymbol{\varphi}_l^2$. \mathbf{s}_i is defined as

$$\mathbf{s}_i = \frac{\lambda}{1 + \text{dist}(\boldsymbol{\varphi}_k^1, \boldsymbol{\varphi}_l^2)} \cdot \frac{\text{distNN}_{\boldsymbol{\varphi}_k^1}^{2\text{nd}}}{\text{distNN}_{\boldsymbol{\varphi}_k^1}^{1\text{st}}}, \quad (11)$$

where $\text{distNN}^{1\text{st}}$ and $\text{distNN}^{2\text{nd}}$ denote Euclidean distance between ϕ_k^1 and its first and second nearest neighbour in descriptor space. λ is a scale factor that we set to 0.03 manually. The ratio between first and second nearest neighbour can represent the distinctiveness of detected keypoints and thus is used to reduce the errors that introduced repeated patterns.

Similar as c_i , high s_i means more contribution of ϵ_i to optimization problem as is shown in (10). The higher s_i is, the more difficult for ϵ_i to change which means the less likely that c_i deviates from its expected value \bar{c}_i which is 1. In other words, the higher the s_i is, the more likely that m_i is classified as an inlier.

The reason why we choose to work with descriptor similarity instead of appearance difference is twofold: first, the descriptors and distances to nearest neighbours have been computed before the optimization and thus saving computational time. Second, in this way, \mathbf{S} will stay constant during optimization process, no matter how \mathbf{H} and \mathbf{C} change, because descriptors have been extracted beforehand. Hence, no more new parameters are needed to be optimized.

After introducing all variables of our cost function as in (8), we can have a better intuitive understanding about it. δ_i is responsible for minimizing the geometric error of homography transformation. e_i in it represents how inconsistent the correspondence m_i with estimated homography \mathbf{H} . In order to minimize it, high e_i which normally means outliers, will push c_i towards zero. Although it has the nice property of ignoring the errors introduced by outliers, even for low e_i which is nonzero, nothing prevents c_i reducing its value. This means that effects of all m_i are reduced even ignored during optimization which will result in wrong estimation of homography. Fortunately, ρ_i helps to avoid this behaviour. It serves as a penalty whenever c_i decreases. This is achieved by setting all initial value of c_i and its expected value \bar{c}_i to 1. As a result, decreasing c_i corresponds to increasing prior error ϵ_i . Moreover, we introduce s_i to reflect similarity and distinctiveness of detected keypoints in m_i and the value of it determines the amount of penalty added to the cost function.

The least squares optimization can be further robustified. Due to Gaussian assumption, the error vectors e_i and ϵ_i in (8) have quadratic influence on cost function $F'(x)$. Even a single outlier would have major negative effect. In order to be a more outlier robust, the Huber cost function is used and makes cost increase linearly if error exceeds certain threshold; thus (8) becomes

$$F'(x) = \begin{cases} F'(x), & \text{if } \sqrt{F'(x)} < b \\ 2b\sqrt{F'(x)} - b^2, & \text{otherwise.} \end{cases} \quad (12)$$

There are even more robust cost functions such as Blake-Zisserman, corrupted Gaussian, or Cauchy. Compared to them, the reason why Huber function is preferred is that it does not introduce new local minima due to its convexity [13].

For a given optimization problem, there are often several ways to parametrize it. For each c_i , one parameter is enough to parametrize it since it is a weight factor for the contribution of δ_i to optimization problem. Although \mathbf{H} has only 8 DOF, we parametrize it using 9 parameters. As discussed in [13], it is not necessary or advisable to use 8-parametrization

by removing scale factor because when using the minimal parameterization, it is more likely for the optimization process stuck in the local minimum. Also, in our problem, the number of parameters in \mathbf{C} is usually much more than 9. A minimal parameterization of \mathbf{H} has little help on efficiency of our approach. Hence, the total number of parameters need to be solved during optimization is $9+n$ where n is the number of keypoint correspondences.

3. Experiments and Evaluations

In this section we evaluate our approach using both synthetic and real-life pictures. Synthetic images are generated by transforming the original image I_s to target image I_t through a homography \mathbf{H} generated randomly. We apply FAST [14] keypoint detector on I_s and I_t . BRIEF [15] descriptors are extracted from detected keypoints. Keypoint correspondences \mathbf{M} are determined simply by finding the nearest neighbours in descriptor space.

3.1. Synthetic Noise. To evaluate the performance of our approach, besides the original outliers, more randomly generated false correspondences are added to \mathbf{M} . The homography estimation results of our approach are shown in Table 1. The accuracy of estimated homography is evaluated using root mean-squared error (RMSE). From the results we can see that even the number of outliers is over ten times more than inliers; our homography estimation still remains feasible compared to the approach of minimizing the conventional cost function (RMSE w/o \mathbf{C}).

Our purpose to introduce \mathbf{S} into cost function is to integrate descriptor similarity into optimization process and penalize the behaviour of decreasing \mathbf{C} . The lower the s_i is, the easier it is to decrease c_i and thus the easier the m_i is classified as an outlier. The column "RMSE w/o \mathbf{S} " in Table 1 shows that without \mathbf{S} , the algorithm still has some robustness against outliers even when the error rate is higher than 0.5; however, the RMSE increases significantly especially when the error rate goes higher.

When the optimization process finishes, a keypoint correspondence is classified as outlier if its c_i is below 0.5, since we initially set c_i to 1 and its value tends to move towards 0. False inliers deviate homography estimation more severely than false outliers because using a wrong correspondence during optimization process is much worse than not using a correct correspondence. The proposed approach can keep the number of false inliers very low compared to the total number of correspondences. Because many outliers are generated randomly, some of them are accidentally close to the truth as shown in Figure 1. Such false matchings are acceptable to our algorithm. Because the error it introduced to the whole optimization problem is considerably small and any effort to reject such outliers would introduce new parameters to be tuned and thus complicate the problem.

It is worth noticing that more outliers do not necessarily mean more error on homography estimation. For example, in Table 1, the RMSE of the test with 309 outliers is lower than many of those with fewer outliers. The reason for this

TABLE 1: Homography estimation results on random generated false matches. TP and FP mean true positive and false positive number, while TN and FN represent true negative and false negative number. The column “RMSE w/o \mathbf{S} ” represents the RMSE of approach without using descriptor similarity \mathbf{S} , while “RMSE w/o \mathbf{C} ” represents the RMSE of standard optimization approach whose cost function is introduced in (6).

True matches	False matches	TP	FP	TN	FN	RMSE	RMSE w/o \mathbf{S}	RMSE w/o \mathbf{C}
42	0	42	0	0	0	0.166	0.165	0.184
42	51	42	1	50	0	0.573	0.462	13.992
42	103	42	1	102	0	0.597	0.699	21.420
42	154	42	1	153	0	0.222	0.608	27.589
42	206	42	1	205	0	0.633	5.816	30.845
42	257	42	1	256	0	0.622	17.311	34.856
42	309	42	1	308	0	0.228	28.479	37.672
42	360	41	1	359	1	0.652	36.286	38.408
42	412	42	1	411	0	0.594	37.278	40.348
42	463	41	1	462	1	0.821	39.889	41.875
42	515	42	1	514	0	0.825	38.737	42.467

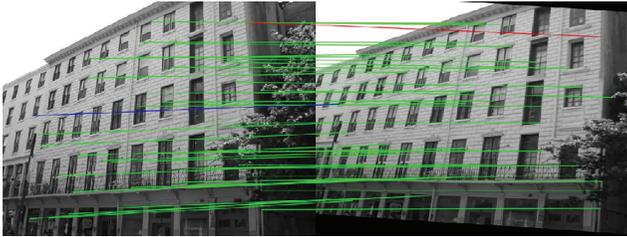


FIGURE 1: Keyframe correspondences between original and transformed images with 567 outliers. The green lines indicate correctly detected inliers, while blue means false negative and red means false positive detections. The correctly detected outliers are ignored for the sake of clearance of the figure.

is that almost all of the \mathbf{c}_i lies in the area that is either very close to 1 or very close to 0 as seen in Figure 2(b). This means the effects of most outliers are removed instead of reduced, while, in the case of 206 outliers as shown in Figure 2(a), \mathbf{c}_i of some correspondences lies close to 0.5 which means that the effect of some inliers is not enforced properly. Hence, even tests with fewer total outliers may still have higher RMSE.

The best case is to assign \mathbf{c}_i of detected outliers to 0. However, this cannot be done naturally in our algorithm. In order to minimize (8), both \mathbf{H} and \mathbf{C} need to be optimized. As their Jacobians are needed to solve the optimization problem using an iterative method, the values of \mathbf{C} during the optimization process have to be continuous instead of discrete. Hence, this inaccuracy is inevitable by just solving cost function. However, it is very easy to refine the homography because our algorithm naturally assigns each matching a confidence \mathbf{c}_i which is based on both the geometric error and descriptor similarity. We simply select the best m inliers to refine the homography using DLT; then the inaccuracy caused by continuous \mathbf{C} can be removed.

We compare our approach with RANSAC and LMedS. Homography is estimated using the same data set as in Table 1 and the result is shown in Figure 3(a).

As expected, the RMSE of LMedS rise when error rate is over 50%. With our robustified problem formulation, the RMSE of our approach keeps low and almost constant with the increasing number of outliers. Our approach shows similar robustness compared to RANSAC and the same robustness can be easily achieved by using the refinement described above. We use optimized OpenCV implementations for both RANSAC and LMedS. We repeated tests 5 times for 11 noise levels and the run time is averaged for each level. The result is shown in Figure 3(b). LMedS approach keeps low and constant runtime at around 4 ms. The runtime for our approach started at 3.3 ms with 42 inliers and 0 outliers ended with 46.08 ms with 567 outliers. The RANSAC approach started at 0.81 ms and ended with 18249 ms. The runtime of RANSAC increases much faster than our approach because the required number of sampling times increases significantly with the portion of outliers [13].

The problem size of our approach increases with the number of keypoint correspondences. Besides 9-parameter homography, for each new correspondence \mathbf{m}_i , a new parameter \mathbf{c}_i that needs to be optimized is added to the problem formulation. Jacobian \mathbf{J} and Hessian ($\mathbf{J}^T\mathbf{J}$) are required to minimize the cost function and their size also increases with the number of parameters. When solving $n \times n$ equations, it takes $O(n^3)$ for a dense system [16]. Thus, the size of the problem is not the bottle neck as solving it is much more crucial.

By exploiting the sparseness of the optimization problem, computational benefits can be gained through avoiding storing and operating on zero elements [17]. Hence, the sparser the structure, the more computational benefits we can gain.

Fortunately, when solving our minimization problems, the Jacobian and Hessian matrix in our problem have a sparse block structure as shown in Figure 4. Only the black entries are nonzero. We can divide the Jacobian matrix into 4 zones. Top-left zone is the Jacobian for geometric error \mathbf{e} with respect to homography \mathbf{H} . The geometric error for each correspondence \mathbf{e} is a $2d$ vector and the number of

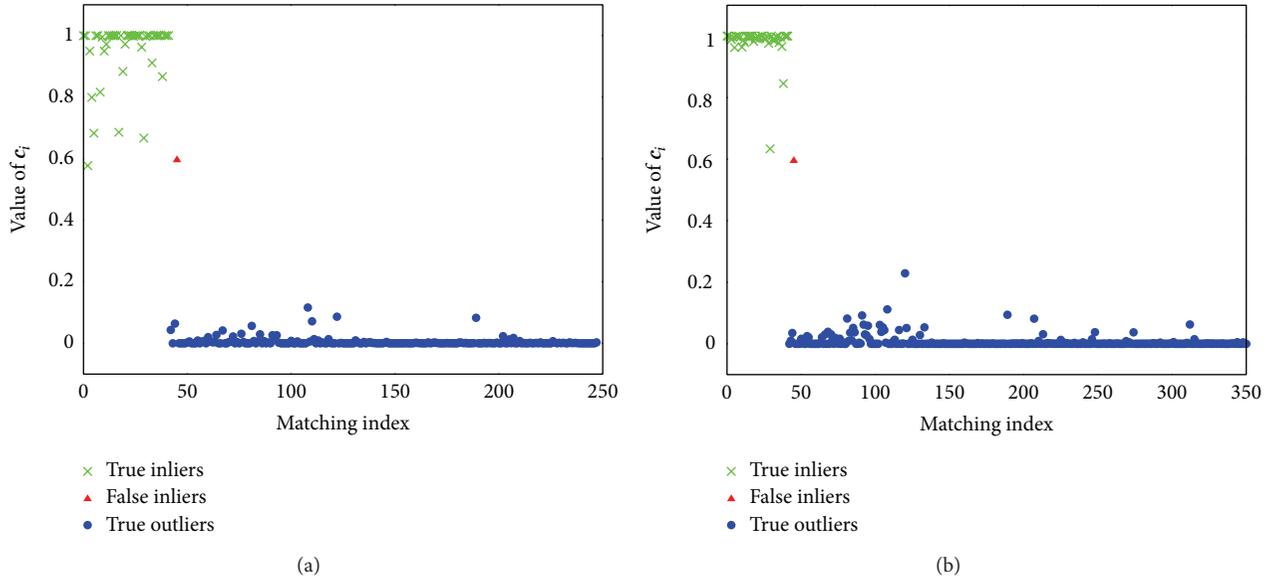


FIGURE 2: Distribution of C which represents the contribution of each matching to the optimization problem. Also, it is an indicator of outliers as we classify m_i as outlier if $c_i < 0.5$. Without loss of generality, inliers are put in the first 42 indexes. (a) Test with 206 outliers in which 1 false inlier appeared. (b) Test with 309 outliers in which 1 false inlier appeared.

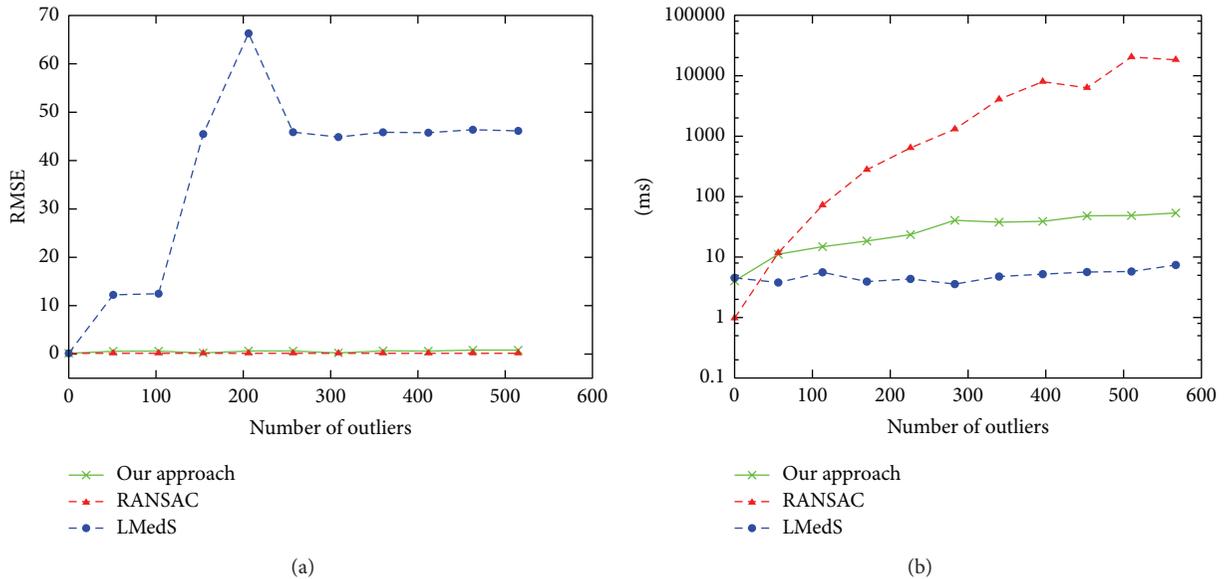


FIGURE 3: Performance comparison with RANSAC and LMedS. (a) Root mean squared error comparison and (b) runtime comparison.

parameters of homography is 9. The top-right zone is the partial derivatives of \mathbf{e} with respect to parameters C whose size equals the number of correspondences. Because c_i only influences \mathbf{e}_i , the only nonzero entries in this area are $\partial \mathbf{e}_i / \partial c_i$. The entries in bottom-left zone are partial derivatives of \mathbf{e} with respect to \mathbf{H} and they are all zero. Because we only use descriptor similarity whose value is independent on homography estimation to prevent that \mathbf{H} is involved into calculation of \mathbf{e} . The bottom-right zone is diagonal and the only nonzero entries are the partial derivatives $\partial \mathbf{e}_i / \partial c_i$.

It is easy to calculate that the density of Jacobian matrix is $5/(9+n)$, while for Hessian it is $(63+19n)/(9+n)^2$. Hence the more the number of correspondences is, the sparser the matrix would be which means more computational benefits can be obtained as we only operate on nonzero blocks.

The convergence property of proposed approach is depicted in Figure 5. For homography estimation, compared to the standard cost function which reports geometric error only, our approach introduces additional parameters contributing to the overall error. As can be seen from the figure,

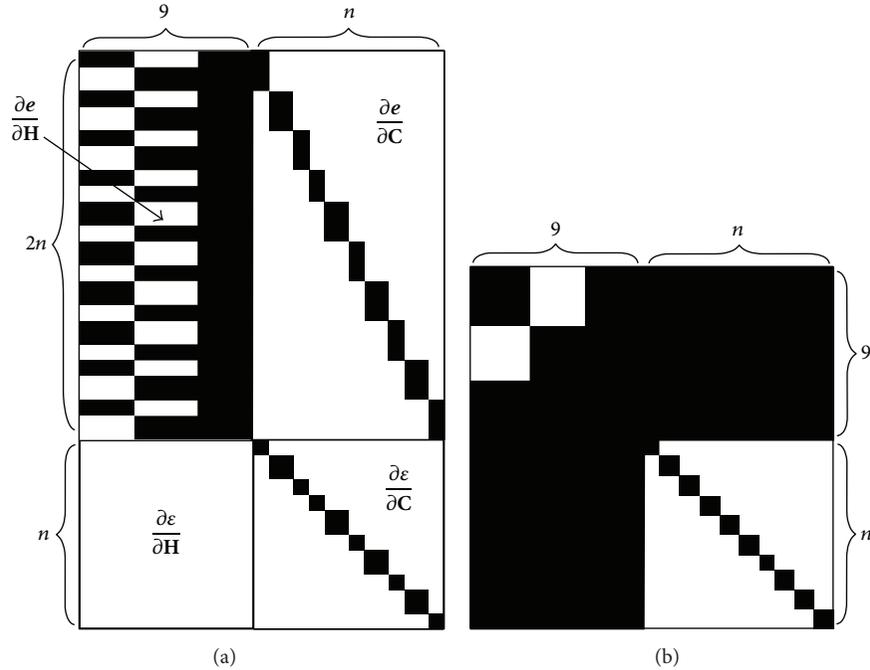


FIGURE 4: Sparse structure example of Jacobian \mathbf{J} (a) and Hessian $\mathbf{J}^T \mathbf{J}$ (b) of the proposed optimization problem. The black entries are nonzero. n is the number of keypoint correspondences. In this specific example, n equals 10.

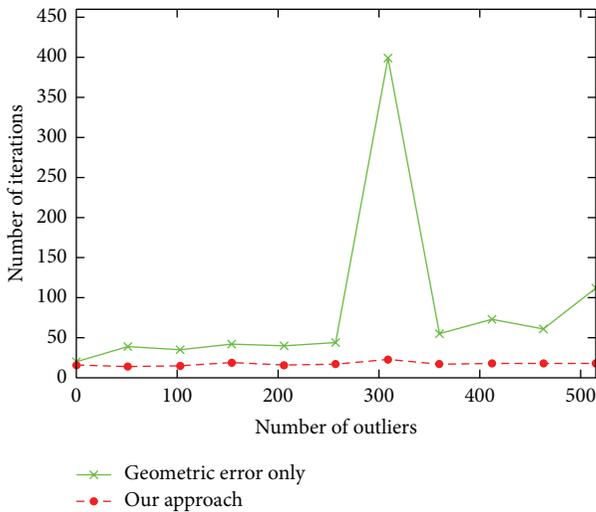


FIGURE 5: Number of iterations needed before optimization process converges. For our approach, the number of iterations is between 14 and 23.

our approach needs about 15 iterations to converge, while the standard cost function for homography estimation needs significantly more iterations to be optimized. This indicates that compared to the standard cost function, our cost function has fewer local minimals and the newly introduced parameters offer faster convergence during optimization process.

3.2. *Experiments on Real-Life Images.* Compared to synthetic images, homography estimation is more complex for real-life images due to illumination variation or camera distortions and so on. The robustness and efficiency of our approach are tested using real-life images and the results are shown in Table 2. The input and reference image pairs are under different viewing conditions. To evaluate the homography estimation, we transform the input image through the estimated homography and the differences between reference images and transformed images are shown in the third and fourth columns in Table 2. We can see that the result is similar as the one in the synthesized experiment. Our approach can achieve similar robustness as RANSAC while saving much in computational time.

4. Conclusion

In this paper, we propose a novel homography estimation and outlier detection approach which is essentially different from conventional hypothesis-and-verify approaches such as RANSAC. We formulate the homography estimation and outlier detection problem together into a single nonlinear least squares problem. The new cost function combines both geometric error and descriptor similarity, by minimizing which of the homography and outliers can be determined simultaneously.

Experiment results demonstrate that our approach achieves similar robustness as RANSAC under different viewing conditions. Due to the sparse structure of Jacobian and

TABLE 2: Experiment results on real pictures. The first pair of images is under different blur levels and the second pair with different orientations and zoom. The third pair is for different view angles and repeated patterns.

Input	Reference	Our approach	RANSAC	Time (ours)	Time (RANSAC)
				16.79 ms	190.94 ms
				13.08 ms	4525.1 ms
				18 ms	57 ms

Hessian of proposed cost function, our algorithm remains efficient even with the presence of large amount of outliers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.
- [2] D. Nister, "Preemptive ransac for live structure and motion estimation," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 199–206, 2003.
- [3] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [4] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '08)*, pp. 125–134, 2008.
- [5] Y. Ke and R. Sukthankar, "Pca-sift: a more distinctive representation for local image descriptors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, pp. II-506–II-513, 2004.
- [6] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [7] O. Chum and J. Matas, "Matching with prosac—progressive sample consensus," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 220–226, 2005.
- [8] R. Raguram, J. M. Frahm, and M. Pollefeys, "A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus," in *European Conference on Computer Vision*, vol. 5303, pp. 500–513, 2008.
- [9] P. Torr and A. Zisserman, "Mlesac: a new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.
- [10] P. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *International Journal of Computer Vision*, vol. 50, no. 1, pp. 35–61, 2002.
- [11] P. J. Huber, *Robust Statistics*, John Wiley & Sons, 1981.
- [12] E. Serradell, M. Özuysal, V. Lepetit, P. Fua, and F. Moreno-Noguer, "Combining geometric and appearance priors for robust homography estimation," in *European Conference on Computer Vision*, vol. 6313, pp. 58–72, 2010.
- [13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2004.
- [14] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, vol. 3951, pp. 430–443, 2006.
- [15] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 1281–1298, 2012.
- [16] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Vision Algorithms: Theory and Practice*, Lecture Notes in Computer Science, pp. 298–375, Springer, 2000.
- [17] M. I. A. Lourakis and A. A. Argyros, "Sba: a software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software*, vol. 36, pp. 1–30, 2009.

Research Article

Neighborhood Preserving Convex Nonnegative Matrix Factorization

Jiang Wei, Li Min, and Zhang Yongqing

School of Mathematics, Liaoning Normal University, Dalian 116029, China

Correspondence should be addressed to Jiang Wei; swxxjw@aliyun.com

Received 27 September 2013; Accepted 6 January 2014; Published 25 February 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Jiang Wei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The convex nonnegative matrix factorization (CNMF) is a variation of nonnegative matrix factorization (NMF) in which each cluster is expressed by a linear combination of the data points and each data point is represented by a linear combination of the cluster centers. When there exists nonlinearity in the manifold structure, both NMF and CNMF are incapable of characterizing the geometric structure of the data. This paper introduces a neighborhood preserving convex nonnegative matrix factorization (NPCNMF), which imposes an additional constraint on CNMF that each data point can be represented as a linear combination of its neighbors. Thus our method is able to reap the benefits of both nonnegative data factorization and the purpose of manifold structure. An efficient multiplicative updating procedure is produced, and its convergence is guaranteed theoretically. The feasibility and effectiveness of NPCNMF are verified on several standard data sets with promising results.

1. Introduction

This nonnegative matrix factorization (NMF) [1, 2] has been widely used in information retrieval, computer vision, pattern recognition, and DNA gene expressions [3, 4]. NMF decomposes the data matrix as the product of two matrices that possess only nonnegative elements. It has been stated by many researchers that there are a lot of favorable properties for such a decomposition over other similar decompositions, such as PCA. One of the most useful properties of NMF is that it usually leads to parts-based representation because it allows only additive, not subtractive, combinations. Such a representation encodes much of the data making them easy to interpret. NMF can be traced back to 1970s and has been studied extensively by Paatero and Tapper [5]. The work of Lee and Seung [1] brought much attention to NMF in machine learning and data mining fields. Since then, various extensions and variations of NMF have been proposed. Li et al. [4] proposed local nonnegative matrix factorization (LNMF) algorithm which imposes extra constraints to the cost function to get more localized and parts-based image features. Hoyer [6, 7] employed sparsity constraints to improve local data representation, while nonnegative tensor

factorization was studied in [8, 9] by Hazan et al. to handle the data encoded as high-order tensors. All the methods mentioned above are unsupervised, Wang et al; [10] and Zafeiriou et al. [11] proposed independently the Fisher-NMF, which was further studied by Kotsia et al. [12], by adding an additional constraint seeking to maximize the between-class scatter and minimize the within-class scatter in the subspace spanned by the bases.

One of the most important drawbacks of NMF and its variants is the fact that these methods have to be performed in the original feature space of the data points, so that it can not be kernelized and the powerful idea of the kernel method cannot be applied to NMF. Ding et al. [13] proposed convex nonnegative factorization (CNMF) that strives to address the problems while inheriting all the strengths of the above NMF method, which models each cluster as a linear combination of the data points and each data point as a linear combination of the cluster centers. The major advantage of CNMF over NMF is that it can be performed on any data representations, either in the original space or RKHS.

Recently, there has been a lot of interest in geometrically motivated approaches to data analysis in high dimensional spaces. When the data lives on or close to a nonlinear

low dimensional manifold which is embedded in the high dimensional ambient space [14, 15], Euclidean distance is incapable of charactering the geometric structure of the data and hence traditional methods like NMF and CNMF no longer work well. Both CNMF and NMF do not exploit the geometric structure of the data, which assume that the data points are sampled from a Euclidean space. To address this problem, Cai et al. proposed a graph regularized NMF (GNMF) [16] and locally consistent concept factorization (LCCF) [17], which assume that the nearby data points are likely to be in the same cluster, that is, cluster assumption [18, 19]. The Euclidean and manifold geometry are unified through a regularization framework, which has a better interpretation from manifold perspective.

In this paper, we introduce a novel matrix factorization algorithm, called neighborhood preserving convex nonnegative matrix factorization (NPCNMF) which is based on the assumption that if a data point can be reconstructed from its neighbors in the input space, then it can be reconstructed from its neighbors by the same reconstruction coefficients in the low dimensional subspace, that is, local linear embedding assumption [20]. NPCNMF not only inherits the advantages of CNMF, for example, non-negativity, but also overcomes the shortcomings of CNMF, that is, Euclidean assumption. We also propose a multiplicative algorithm to efficiently solve the corresponding optimization problem and its convergence is theoretically guaranteed.

The rest of this paper is organized as follows. In Section 2, we briefly review NMF and CNMF. In Section 3, we introduce our algorithm and provide the proof of the convergence of the proposed algorithm. Experiments on three benchmark face recognition data sets are demonstrated in Section 4. Finally, we draw a conclusion and provide suggestions for future work.

2. A Review of NMF and CNMF

Nonnegative matrix factorization (NMF) factorizes the data matrix into one nonnegative basis matrix and one nonnegative coefficient matrix. Given a nonnegative data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}_+^{m \times N}$, each column of \mathbf{X} is a sample point. NMF aims to find two nonnegative matrices $\mathbf{U} \in \mathbb{R}_+^{m \times r}$ and $\mathbf{V} \in \mathbb{R}_+^{N \times r}$ which minimize the following objective function:

$$\mathcal{J}_{\text{NMF}} = \|\mathbf{X} - \mathbf{UV}^T\|^2 \quad \text{s.t. } \mathbf{U} \geq 0, \mathbf{V} \geq 0, \quad (1)$$

where $\|\cdot\|_F$ is Frobenius norm.

The objective function is joint optimization problem of basis matrix \mathbf{U} and coefficient matrix \mathbf{V} . Although it is not jointly convex to \mathbf{U} and \mathbf{V} , it is convex with respect to each of them when the other one is fixed. Therefore, it is unrealistic to expect an algorithm to find the global minimum of \mathcal{J}_{NMF} .

To optimize the objective, Lee and Seung [2] presented an iterative multiplicative updating algorithm as follows:

$$\begin{aligned} u_{ik}^{t+1} &= u_{ik}^t \frac{(\mathbf{XV})_{ik}}{(\mathbf{UV}^T\mathbf{V})_{ik}}, \\ v_{jk}^{t+1} &= v_{jk}^t \frac{(\mathbf{X}^T\mathbf{U})_{jk}}{(\mathbf{VU}^T\mathbf{U})_{jk}}. \end{aligned} \quad (2)$$

It is proved that the above updated steps will find a local minimum of the objective function in (1).

In reality, we have $r \ll m$ and $r \ll N$. Thus, NMF essentially tries to find a compressed approximation of the original matrix, $\mathbf{X} \approx \mathbf{UV}^T$. We can view this approximation column as follows:

$$\mathbf{x}_j = \sum_{k=1}^r \mathbf{u}_k v_{jk}, \quad (3)$$

where \mathbf{u}_k is the k th column vector of \mathbf{U} . Thus, each data vector \mathbf{x}_j is approximated by a linear combination of the columns of \mathbf{U} , weighted by the components of \mathbf{V} . One limitation of NMF is that the nonnegative requirement is not applicable to applications where the data involves negative number. The second is that it is not clear how to effectively perform NMF in the transformed data space so that the powerful kernel method can be applied. To overcome the problem, Ding et al. [13] proposed a convex nonnegative matrix factorization (CNMF) algorithm where nonnegative and mixed-sign data matrices are applied. CNMF claims that each base can be characterized by a linear combination of the entire data points while each data point can be approximated by a linear combination of all the bases. Translating the statements into mathematics, we have

$$\mathbf{u}_k = \sum_{j=1}^N \mathbf{x}_j w_{jk}, \quad (4)$$

$$\mathbf{x}_j = \sum_{k=1}^r \mathbf{u}_k v_{jk}, \quad (5)$$

where w_{jk} is a nonnegative weight in which data point \mathbf{x}_j is related to k th base and v_{jk} is a nonnegative projection value of \mathbf{x}_j . Replacing \mathbf{u}_k in (5) with (4), we have

$$\mathbf{x}_j = \sum_{k=1}^r \sum_{j=1}^N \mathbf{x}_j w_{jk} v_{jk}. \quad (6)$$

We form the $m \times N$ data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ using the feature vector of data point \mathbf{x}_i as the i th column, the $N \times r$ matrix $\mathbf{W} = [w_{ij}]$ using bases w_{ij} , and $N \times r$ projection matrix $\mathbf{V} = [v_{ij}]$ using the projection values v_{ij} . From (6), we have

$$\mathbf{X} = \mathbf{XWV}^T. \quad (7)$$

Equation (7) can be interpreted as the approximation of the original data set. Minimizing the squared error \mathcal{J} and its approximation [13]

$$\mathcal{J}_{\text{CNMF}} = \|\mathbf{X} - \mathbf{XWV}^T\|^2, \quad (8)$$

where $\mathbf{X} \in \mathbb{R}^{m \times N}$, $\mathbf{W} \in \mathbb{R}^{N \times r}$, $\mathbf{V} \in \mathbb{R}^{N \times r}$. The matrices \mathbf{W} and \mathbf{V} are updated iteratively until convergence using the following rules:

$$\begin{aligned} w_{ik} &= w_{ik} \sqrt{\frac{(\mathbf{Y}^+ \mathbf{V})_{ik} + (\mathbf{Y}^- \mathbf{W} \mathbf{V}^T \mathbf{V})_{ik}}{(\mathbf{Y}^- \mathbf{V})_{ik} + (\mathbf{Y}^+ \mathbf{W} \mathbf{V}^T \mathbf{V})_{ik}}}, \\ v_{ik} &= v_{ik} \sqrt{\frac{(\mathbf{Y}^+ \mathbf{W})_{ik} + (\mathbf{V} \mathbf{W}^T \mathbf{Y}^- \mathbf{W})_{ik}}{(\mathbf{Y}^- \mathbf{W})_{ik} + (\mathbf{V} \mathbf{W}^T \mathbf{Y}^+ \mathbf{W})_{ik}}}, \end{aligned} \quad (9)$$

where $\mathbf{Y} = \mathbf{X}^T \mathbf{X}$ and the matrix \mathbf{Y}^+ and \mathbf{Y}^- are given by

$$\begin{aligned} \mathbf{Y}_{ik}^+ &= \frac{1}{2} (|\mathbf{Y}_{ik}| + \mathbf{Y}_{ik}), \\ \mathbf{Y}_{ik}^- &= \frac{1}{2} (|\mathbf{Y}_{ik}| - \mathbf{Y}_{ik}), \end{aligned} \quad (10)$$

respectively.

3. Neighborhood Preserving Convex Nonnegative Matrix Factorization

In this section, we introduce our neighborhood preserving convex nonnegative matrix factorization method, which takes the local linear embedding constraint as an additional requirement. The method presented in this paper is fundamentally motivated from the neighborhood preserving embedding.

3.1. The Objective Function. Many real world data are actually sampled from a nonlinear low dimensional manifold which is embedded in the high dimensional ambient space. Both NMF and CF perform the factorization in the Euclidean space. They fail to discover the local geometrical structure of the data space, which is essential to the clustering problem. NPE aims at preserving the local manifold structure. Specifically, for each data point, it is represented as a linear combination of the neighboring data points and the combination coefficients are specified in the weight matrix. We can find an optimal embedding such that the combination coefficients can be preserved in the low dimensional subspace.

For each data point, we find its k nearest neighbors. And we can characterize the local geometrical structure by linear coefficients that reconstruct each data point from its neighbors. The reconstruction coefficients are computed by the following objective function:

$$\begin{aligned} \min \quad & \left\| \mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)} w_{ij} \mathbf{x}_j \right\|^2 \\ \text{s.t.} \quad & \sum_{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)} w_{ij} = 1, \end{aligned} \quad (11)$$

and $w_{ij} = 0$ if $\mathbf{x}_j \notin \mathcal{N}_k(\mathbf{x}_i)$, where $\mathcal{N}_k(\mathbf{x}_i)$ denotes the k nearest neighborhood of \mathbf{x}_i .

Then \mathbf{v}_i , $1 \leq i \leq n$ in the dimensionality reduced space can be preserved by minimizing

$$\begin{aligned} \min \left\| \mathbf{v}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)} w_{ij} \mathbf{v}_j \right\|^2 &= \text{tr}(\mathbf{V}^T (\mathbf{I} - \mathbf{W}) (\mathbf{I} - \mathbf{W}) \mathbf{V}) \\ &= \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}), \end{aligned} \quad (12)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, $\mathbf{I} \in \mathbb{R}^{n \times n}$ is an identity matrix, and $\mathbf{L} = (\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W})$. By minimizing (12), we essentially try to formalize our intuition that if a data point can be represented from its neighbors in the original space, then it can be represented from its neighbors by the same combination coefficients in the dimensionality reduced space.

With the neighborhood preserving constraint, CNMF incorporates (8) and minimizes the objective function as follows:

$$\mathcal{J} = \left\| \mathbf{X} - \mathbf{X} \mathbf{W} \mathbf{V}^T \right\|^2 + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}), \quad (13)$$

where λ is a positive regularization parameter controlling the contribution of the additional constraint. We call (13) neighborhood preserving convex nonnegative matrix factorization (NPCNMF). Let $\lambda = 0$; (13) degenerates to the original CNMF.

3.2. The Algorithm. We introduce an iterative algorithm to find a local minimum for the optimization problem. By defining $\mathbf{K} = \mathbf{X}^T \mathbf{X}$ and using the matrix properties $\|\mathbf{A}\|^2 = \text{tr}(\mathbf{A}^T \mathbf{A})$, $\text{tr}(\mathbf{A} \mathbf{B}) = \text{tr}(\mathbf{B} \mathbf{A})$, and $\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{A}^T)$, we can rewrite the objective function \mathcal{J} as follows:

$$\begin{aligned} \mathcal{J} &= \text{tr} \left((\mathbf{X} - \mathbf{X} \mathbf{W} \mathbf{V}^T)^T (\mathbf{X} - \mathbf{X} \mathbf{W} \mathbf{V}^T) \right) + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \\ &= \text{tr} \left((\mathbf{I} - \mathbf{W} \mathbf{V}^T)^T \mathbf{K} (\mathbf{I} - \mathbf{W} \mathbf{V}^T) \right) + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \\ &= \text{tr}(\mathbf{K}) - 2 \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K}) + \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K} \mathbf{W} \mathbf{V}^T) \\ &\quad + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}). \end{aligned} \quad (14)$$

This is a typical constrained optimization problem and can be solved using the Lagrange multiplier method. Let ψ_{ij} and ϕ_{ij} be the Lagrange multiplier for constraint $w_{ij} \geq 0$ and $v_{ij} \geq 0$, respectively, and let $\Psi = [\psi_{ij}]$ and $\Phi = [\phi_{ij}]$. The Lagrangian function is

$$\begin{aligned} \mathcal{L} &= \text{tr}(\mathbf{K}) - 2 \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K}) + \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K} \mathbf{W} \mathbf{V}^T) \\ &\quad + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \text{tr}(\Psi \mathbf{W}^T) + \text{tr}(\Phi \mathbf{V}^T). \end{aligned} \quad (15)$$

The partial derivatives of \mathcal{L} with respect to \mathbf{W} and \mathbf{V} are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = -2 \mathbf{K} \mathbf{V} + 2 \mathbf{K} \mathbf{W} \mathbf{V}^T \mathbf{V} + \Psi, \quad (16)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = -2 \mathbf{K} \mathbf{W} + 2 \mathbf{V} \mathbf{W}^T \mathbf{K} \mathbf{W} + 2 \lambda \mathbf{L} \mathbf{V} + \Phi.$$

Using the Karush-Kuhn-Tucker conditions $\psi_{ik}w_{ik} = 0$ and $\phi_{jk}v_{jk} = 0$, we get the following equations for w_{ik} and v_{jk} :

$$\begin{aligned} (-\mathbf{KV} + \mathbf{KWV}^T\mathbf{V})_{ik}w_{ik} &= 0, \\ (-\mathbf{KW} + \mathbf{VW}^T\mathbf{KW} + \lambda\mathbf{LV})_{jk}v_{jk} &= 0. \end{aligned} \quad (17)$$

The corresponding equivalent formulas are as follows:

$$\begin{aligned} (-\mathbf{KV} + \mathbf{KWV}^T\mathbf{V})_{ik}w_{ik}^2 &= 0, \\ (-\mathbf{KW} + \mathbf{VW}^T\mathbf{KW} + \lambda\mathbf{LV})_{jk}v_{jk}^2 &= 0. \end{aligned} \quad (18)$$

Introduce

$$\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-, \quad (19)$$

where $\mathbf{A}_{ik}^+ = (|\mathbf{A}_{ik}| + \mathbf{A}_{ik})/2$ and $\mathbf{A}_{ik}^- = (|\mathbf{A}_{ik}| - \mathbf{A}_{ik})/2$.

The equations lead to the following updating formulas:

$$w_{ik} = w_{ik} \sqrt{\frac{(\mathbf{K}^+\mathbf{V})_{ik} + (\mathbf{K}^-\mathbf{VW}^T\mathbf{V})_{ik}}{(\mathbf{K}^-\mathbf{V})_{ik} + (\mathbf{K}^+\mathbf{VW}^T\mathbf{V})_{ik}}}, \quad (20)$$

$$v_{jk} = v_{jk} \sqrt{\frac{(\mathbf{K}^+\mathbf{W})_{jk} + (\mathbf{VW}^T\mathbf{K}^-\mathbf{W})_{jk} + \lambda(\mathbf{L}^-\mathbf{V})_{jk}}{(\mathbf{K}^-\mathbf{W})_{jk} + (\mathbf{VW}^T\mathbf{K}^+\mathbf{W})_{jk} + \lambda(\mathbf{L}^+\mathbf{V})_{jk}}}. \quad (21)$$

Note that the solution to minimizing the criterion function \mathcal{F} is not unique. If \mathbf{W} and \mathbf{V} are the solution, then, $\mathbf{W}\mathbf{D}$, $\mathbf{W}\mathbf{D}^{-1}$ will also form a solution for any positive diagonal matrix \mathbf{D} . To make the solution unique, we will further require that $\mathbf{w}^T\mathbf{K}\mathbf{w} = 1$, where \mathbf{w} is the column vector of \mathbf{W} . The matrix \mathbf{V} will be adjusted accordingly so that \mathbf{WV}^T does not change. This can be achieved by

$$\begin{aligned} \mathbf{V} &\leftarrow \mathbf{V}[\text{diag}(\mathbf{W}^T\mathbf{KW})]^{1/2}, \\ \mathbf{W} &\leftarrow \mathbf{W}[\text{diag}(\mathbf{W}^T\mathbf{KW})]^{1/2}. \end{aligned} \quad (22)$$

3.3. Convergence Analysis. In this section, we will investigate the convergence of the updating formula in (14). We use the auxiliary function approach [16] to prove the convergence. Here we first introduce the definition of auxiliary function [16].

Definition 1. $Z(h, h')$ is an auxiliary function of $F(h)$ if the conditions

$$Z(h, h') \geq F(h), \quad Z(h, h) = F(h) \quad (23)$$

are satisfied.

Lemma 2. If Z is an auxiliary function for F , then F is nonincreasing under the update

$$h^{(t+1)} = \arg \min_h Z(h, h^{(t)}). \quad (24)$$

Proof. Consider

$$F(h^{(t+1)}) \leq Z(h^{(t+1)}, h^{(t)}) \leq Z(h^{(t)}, h^{(t)}) = F(h^{(t)}). \quad (25)$$

□

Lemma 3. For any nonnegative matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{k \times k}$, $\mathbf{S} \in \mathbb{R}^{n \times k}$, and $\mathbf{S}' \in \mathbb{R}^{n \times k}$, and \mathbf{A} , \mathbf{B} are symmetric, then the following inequality holds:

$$\sum_{i=1}^n \sum_{p=1}^k \frac{(\mathbf{A}\mathbf{S}'\mathbf{B})_{ip}^2}{\mathbf{S}'_{ip}} \geq \text{tr}(\mathbf{S}^T\mathbf{A}\mathbf{S}\mathbf{B}). \quad (26)$$

The correctness and convergence of the algorithm are addressed in the following.

For given \mathbf{K} , fixing \mathbf{V} , considering any element w_{ij} in \mathbf{W} , we use $\mathcal{F}(\mathbf{W})$ to denote the part of \mathcal{F} , which is only relevant to w_{ij} . We get

$$\mathcal{F}(\mathbf{W}) = -2 \text{tr}(\mathbf{VW}^T\mathbf{K}) + \text{tr}(\mathbf{VW}^T\mathbf{KWV}^T). \quad (27)$$

Theorem 4. One rewrites $\mathcal{F}(\mathbf{W})$ as follows:

$$\begin{aligned} \mathcal{F}(\mathbf{H}) &= \text{tr}(-2\mathbf{H}^T\mathbf{B}^+ + 2\mathbf{H}^T\mathbf{B}^- + \mathbf{H}^T\mathbf{A}^+\mathbf{H}\mathbf{C} \\ &\quad - \mathbf{H}^T\mathbf{A}^-\mathbf{H}\mathbf{C}), \end{aligned} \quad (28)$$

where $\mathbf{B} = \mathbf{KV}$, $\mathbf{A} = \mathbf{K}$, $\mathbf{C} = \mathbf{V}^T\mathbf{V}$, and $\mathbf{H} = \mathbf{W}$.

Then the following function

$$\begin{aligned} Z(\mathbf{H}, \mathbf{H}') &= -\sum_{ik} 2\mathbf{B}_{ik}^+\mathbf{H}'_{ik} \left(1 + \log \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}}\right) \\ &\quad + \sum_{ik} \mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}^2 + (\mathbf{H}')_{ik}^2}{\mathbf{H}'_{ik}} + \sum_{ik} \frac{(\mathbf{A}^+\mathbf{H}'\mathbf{C})_{ik}\mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}} \\ &\quad - \sum_{ijkl} \mathbf{A}_{ij}^-\mathbf{H}'_{jk}\mathbf{C}_{kl}\mathbf{H}'_{il} \left(1 + \log \frac{\mathbf{H}_{jk}\mathbf{H}_{il}}{\mathbf{H}'_{jk}\mathbf{H}'_{il}}\right) \end{aligned} \quad (29)$$

is an auxiliary function of $\mathcal{F}(\mathbf{H})$: that is, it satisfies the requirements $\mathcal{F}(\mathbf{H}) \leq Z(\mathbf{H}, \mathbf{H}')$ and $\mathcal{F}(\mathbf{H}) = Z(\mathbf{H}, \mathbf{H})$. Furthermore, it is a convex function of \mathbf{H} and its global minimum is

$$\mathbf{H}_{ik} = \underset{\mathbf{H}}{\text{argmin}} \mathbf{H}'_{ik} \sqrt{\frac{\mathbf{B}_{ik}^+ + (\mathbf{H}'\mathbf{A}^-\mathbf{C})_{ik}}{\mathbf{B}_{ik}^- + (\mathbf{A}^+\mathbf{H}'\mathbf{C})_{ik}}}. \quad (30)$$

From its minima and setting $\mathbf{H}^{(t+1)} \leftarrow \mathbf{H}$ and $\mathbf{H}^{(t)} \leftarrow \mathbf{H}'$, one recovers (20), letting $\mathbf{B}^+ = (\mathbf{K}^+)\mathbf{V}$, $\mathbf{B}^- = (\mathbf{K}^-)\mathbf{V}$, $\mathbf{A} = \mathbf{K}$, $\mathbf{C} = \mathbf{V}^T\mathbf{V}$, and $\mathbf{H} = \mathbf{W}$.

Proof. The function $\mathcal{F}(\mathbf{H})$ is

$$\begin{aligned} \mathcal{F}(\mathbf{H}) &= \text{tr}(-2\mathbf{H}^T\mathbf{B}^+ + 2\mathbf{H}^T\mathbf{B}^- + \mathbf{H}^T\mathbf{A}^+\mathbf{H}\mathbf{C} \\ &\quad - \mathbf{H}^T\mathbf{A}^-\mathbf{H}\mathbf{C}). \end{aligned} \quad (31)$$

We find upper bounds for each of the two positive terms and lower bounds for each of the two negative terms. For the third term in $\mathcal{F}(\mathbf{H})$, by applying Lemma 3, we obtain an upper bound

$$\text{tr}(\mathbf{H}^T \mathbf{A}^+ \mathbf{H} \mathbf{C}) \leq \sum_{ik} \frac{(\mathbf{A}^+ \mathbf{H}' \mathbf{C})_{ik} \mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}}. \quad (32)$$

The second term of $\mathcal{F}(\mathbf{H})$ is bounded by

$$\text{tr}(\mathbf{H}^T \mathbf{B}^-) = \sum_{ik} \mathbf{H}_{ik} \mathbf{B}_{ik}^- \leq \sum_{ik} \mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}^2 + (\mathbf{H}')_{ik}^2}{2\mathbf{H}'_{ik}}. \quad (33)$$

To obtain lower bounds for the two remaining terms, we use the inequality $z \geq 1 + \log z$, which holds for any $z > 0$, and the first term in $\mathcal{F}(\mathbf{H})$ is bounded by

$$\text{tr}(\mathbf{H}^T \mathbf{B}^+) = \sum_{ik} \mathbf{B}_{ik}^+ \mathbf{H}_{ik} \geq \sum_{ik} \mathbf{B}_{ik}^+ \mathbf{H}'_{ik} \left(1 + \log \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}}\right). \quad (34)$$

The last term in $\mathcal{F}(\mathbf{H})$ is bounded by

$$\text{tr}(\mathbf{H}^T \mathbf{A}^- \mathbf{H} \mathbf{C}) \geq \sum_{ijkl} \mathbf{A}_{ij}^- \mathbf{H}'_{jk} \mathbf{C}_{kl} \mathbf{H}'_{il} \left(1 + \log \frac{\mathbf{H}_{jk} \mathbf{H}_{il}}{\mathbf{H}'_{jk} \mathbf{H}'_{il}}\right). \quad (35)$$

Collecting all bounds, we obtain $Z(\mathbf{H}, \mathbf{H}')$ as in (29). Obviously, $\mathcal{F}(\mathbf{H}) \leq Z(\mathbf{H}, \mathbf{H}')$ and $\mathcal{F}(\mathbf{H}) = Z(\mathbf{H}, \mathbf{H})$.

To find the minimum of $Z(\mathbf{H}, \mathbf{H}')$, we take

$$\begin{aligned} \frac{\partial Z(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{ik}} &= -2\mathbf{B}_{ik}^+ \frac{\mathbf{H}'_{ik}}{\mathbf{H}_{ik}} + 2\mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}} \\ &+ \frac{2(\mathbf{H}' \mathbf{A}^+ \mathbf{C})_{ik} \mathbf{H}_{ik}}{\mathbf{H}'_{ik}} - \frac{2(\mathbf{H}' \mathbf{A}^- \mathbf{C})_{ik} \mathbf{H}'_{ik}}{\mathbf{H}_{ik}}. \end{aligned} \quad (36)$$

To find the minimum of $Z(\mathbf{H}, \mathbf{H}')$, we take the Hessian matrix of $Z(\mathbf{H}, \mathbf{H}')$

$$\frac{\partial^2 Z(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{ik} \partial \mathbf{H}_{jl}} = \delta_{ij} \delta_{kl} \mathbf{Y}_{ik} \quad (37)$$

To be a diagonal matrix with positive entries

$$\mathbf{Y}_{ik} = \frac{4[(\mathbf{B}^+)_{ik} + (\mathbf{H}' \mathbf{A}^- \mathbf{C})_{ik}] \mathbf{H}'_{ik}}{\mathbf{H}_{ik}^2} + 2 \frac{\mathbf{B}_{ik}^- + (\mathbf{H}' \mathbf{A}^+ \mathbf{C})_{ik}}{\mathbf{H}'_{ik}}. \quad (38)$$

Thus, $Z(\mathbf{H}, \mathbf{H}')$ is a convex function of \mathbf{H} . Therefore, we obtain the global minimum by setting $\partial Z(\mathbf{H}, \mathbf{H}') / \partial \mathbf{H}_{ik} = 0$ in (36) and solving for \mathbf{H} . Rearranging, we obtain (30). \square

Theorem 5. Updating \mathbf{W} using (20) will monotonically decrease the value of the objective in (13); hence it converges.

Proof. By Lemma 2 and Theorem 4, we can get that $\mathcal{F}(\mathbf{W}^t) = Z(\mathbf{W}^{(t)}, \mathbf{W}^{(t)}) \geq Z(\mathbf{W}^{(t+1)}, \mathbf{W}^{(t)}) \geq \mathcal{F}(\mathbf{W}^{(t+1)})$, so $\mathcal{F}(\mathbf{W})$ is monotonically decreasing. Since $\mathcal{F}(\mathbf{W})$ is obviously bounded below, we prove this theorem.

For given \mathbf{K} , fixing \mathbf{W} , considering any element v_{ij} in \mathbf{V} , we use $\mathcal{F}(\mathbf{V})$ to denote the part of \mathcal{F} , which is only relevant to v_{ij} . We get

$$\mathcal{F}(\mathbf{V}) = -2 \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K}) + \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K} \mathbf{W} \mathbf{V}^T) + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}). \quad (39)$$

\square

Theorem 6. One rewrites $\mathcal{F}(\mathbf{V})$ as follows:

$$\begin{aligned} \mathcal{F}(\mathbf{H}) &= \text{tr}(-2\mathbf{H}^T \mathbf{B}^+ + 2\mathbf{H}^T \mathbf{B}^- + \mathbf{H} \mathbf{A}^+ \mathbf{H}^T \\ &- \mathbf{H} \mathbf{A}^- \mathbf{H}^T + \lambda \mathbf{H}^T \mathbf{L}^+ \mathbf{H} - \lambda \mathbf{H}^T \mathbf{L}^- \mathbf{H}), \end{aligned} \quad (40)$$

where $\mathbf{B} = \mathbf{K} \mathbf{W}$, $\mathbf{A} = \mathbf{W}^T \mathbf{X}^T \mathbf{X} \mathbf{W}$, and $\mathbf{H} = \mathbf{V}$.

Then the following function

$$\begin{aligned} Z(\mathbf{H}, \mathbf{H}') &= - \sum_{ik} 2\mathbf{B}_{ik}^+ \mathbf{H}'_{ik} \left(1 + \log \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}}\right) \\ &+ \sum_{ik} \mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}^2 + (\mathbf{H}')_{ik}^2}{\mathbf{H}'_{ik}} + \sum_{ik} \frac{(\mathbf{A}^+ \mathbf{H}')_{ik} \mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}} \\ &- \sum_{ikl} \mathbf{A}_{kl}^- \mathbf{H}'_{ik} \mathbf{H}'_{li} \left(1 + \log \frac{\mathbf{H}_{ik} \mathbf{H}_{li}}{\mathbf{H}'_{ik} \mathbf{H}'_{li}}\right) \\ &+ \lambda \sum_{ik} \frac{(\mathbf{L}^+ \mathbf{H}')_{ik} \mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}} \\ &- \lambda \sum_{ikl} \mathbf{L}_{kl}^- \mathbf{H}'_{ik} \mathbf{H}'_{li} \left(1 + \log \frac{\mathbf{H}_{ik} \mathbf{H}_{li}}{\mathbf{H}'_{ik} \mathbf{H}'_{li}}\right) \end{aligned} \quad (41)$$

is an auxiliary function of $\mathcal{F}(\mathbf{H})$: that is, it satisfies the requirements $\mathcal{F}(\mathbf{H}) \leq Z(\mathbf{H}, \mathbf{H})$ and $\mathcal{F}(\mathbf{H}) = Z(\mathbf{H}, \mathbf{H}')$. Furthermore, it is a convex function of \mathbf{H} and its global minimum is

$$\mathbf{H}_{ik} = \mathbf{H}'_{ik} \sqrt{\frac{\mathbf{B}_{ik}^+ + (\mathbf{H}' \mathbf{A}^-)_{ik} + \lambda (\mathbf{L}^- \mathbf{H}')_{ik}}{\mathbf{B}_{ik}^- + (\mathbf{H}' \mathbf{A}^+)_{ik} + \lambda (\mathbf{L}^+ \mathbf{H}')_{ik}}}. \quad (42)$$

From its minima and setting $\mathbf{H}^{(t+1)} \leftarrow \mathbf{H}$ and $\mathbf{H}^{(t)} \leftarrow \mathbf{H}'$, one recovers (21), letting $\mathbf{B}^+ = (\mathbf{K})^+ \mathbf{U}$, $\mathbf{B}^- = (\mathbf{K})^- \mathbf{U}$, $\mathbf{A} = \mathbf{W}^T \mathbf{K} \mathbf{W}$, and $\mathbf{H} = \mathbf{V}$.

Proof. The function $\mathcal{F}(\mathbf{H})$ is

$$\begin{aligned} \mathcal{F}(\mathbf{H}) &= \text{tr}(-2\mathbf{H}^T \mathbf{B}^+ + 2\mathbf{H}^T \mathbf{B}^- + \mathbf{H} \mathbf{A}^+ \mathbf{H}^T \\ &- \mathbf{H} \mathbf{A}^- \mathbf{H}^T + \lambda \mathbf{H}^T \mathbf{L}^+ \mathbf{H} - \lambda \mathbf{H}^T \mathbf{L}^- \mathbf{H}). \end{aligned} \quad (43)$$

We find upper bounds for each of the three positive terms and lower bounds for each of the three negative terms. For the

third term in $\mathcal{F}(\mathbf{H})$, by applying Lemma 3 and setting $\mathbf{A} \leftarrow \mathbf{I}$, $\mathbf{B} \leftarrow \mathbf{A}^+$, we obtain an upper bound

$$\text{tr}(\mathbf{H}\mathbf{A}^+\mathbf{H}^T) \leq \sum_{ik} \frac{(\mathbf{H}'\mathbf{A}^+)_{ik} \mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}}. \quad (44)$$

The second term of $\mathcal{F}(\mathbf{H})$ is bounded by

$$\text{tr}(\mathbf{H}^T\mathbf{B}^-) = \sum_{ik} \mathbf{H}_{ik}\mathbf{B}_{ik}^- \leq \sum_{ik} \mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}^2 + (\mathbf{H}')_{ik}^2}{2\mathbf{H}'_{ik}}, \quad (45)$$

using the inequality $a \leq (a^2 + b^2)/2b$, which holds for any $a, b > 0$.

For the fifth term in $\mathcal{F}(\mathbf{H})$, setting $\mathbf{A} \leftarrow \mathbf{L}^+$, $\mathbf{B} \leftarrow \mathbf{I}$, and $\mathbf{S} \leftarrow \mathbf{H}$, we obtain an upper bound

$$\lambda \text{tr}(\mathbf{H}^T\mathbf{L}^+\mathbf{H}) \leq \lambda \sum_{ik} \frac{(\mathbf{L}^+\mathbf{H}')_{ik} \mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}}. \quad (46)$$

To obtain lower bounds for the three remaining terms, we use the inequality $z \geq 1 + \log z$, which holds for any $z > 0$, and the first term in $J(\mathbf{H})$ is bounded by

$$\text{tr}(\mathbf{H}^T\mathbf{B}^+) = \sum_{ik} \mathbf{B}_{ik}^+\mathbf{H}_{ik} \geq \sum_{ik} \mathbf{B}_{ik}^+\mathbf{H}'_{ik} \left(1 + \log \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}}\right). \quad (47)$$

The fourth term in $\mathcal{F}(\mathbf{H})$ is bounded by

$$\text{tr}(\mathbf{H}\mathbf{A}^-\mathbf{H}^T) \geq \sum_{ikl} \mathbf{A}_{kl}^-\mathbf{H}'_{ik}\mathbf{H}'_{li} \left(1 + \log \frac{\mathbf{H}_{ik}\mathbf{H}_{li}}{\mathbf{H}'_{ik}\mathbf{H}'_{li}}\right). \quad (48)$$

The last term in $\mathcal{F}(\mathbf{H})$ is bounded by

$$\lambda \text{tr}(\mathbf{H}^T\mathbf{L}^-\mathbf{H}) \geq \lambda \sum_{ikl} \mathbf{L}_{kl}^-\mathbf{H}'_{ik}\mathbf{H}'_{li} \left(1 + \log \frac{\mathbf{H}_{ik}\mathbf{H}_{li}}{\mathbf{H}'_{ik}\mathbf{H}'_{li}}\right). \quad (49)$$

Collecting all bounds, we obtain $Z(\mathbf{H}, \mathbf{H}')$ as in (41). Obviously, $\mathcal{F}(\mathbf{H}) \leq Z(\mathbf{H}, \mathbf{H}')$ and $\mathcal{F}(\mathbf{H}) = Z(\mathbf{H}, \mathbf{H})$.

To find the minimum of $Z(\mathbf{H}, \mathbf{H}')$, we take

$$\begin{aligned} \frac{\partial Z(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{ik}} &= -2\mathbf{B}_{ik}^+ \frac{\mathbf{H}'_{ik}}{\mathbf{H}_{ik}} + 2\mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}} \\ &+ \frac{2(\mathbf{H}'\mathbf{A}^+)_{ik}\mathbf{H}_{ik}}{\mathbf{H}'_{ik}} - \frac{2(\mathbf{H}'\mathbf{A}^-)_{ik}\mathbf{H}'_{ik}}{\mathbf{H}_{ik}} \\ &+ \frac{2\lambda(\mathbf{L}^+\mathbf{H}')_{ik}\mathbf{H}_{ik}}{\mathbf{H}'_{ik}} - \frac{2\lambda(\mathbf{L}^-\mathbf{H}')_{ik}\mathbf{H}'_{ik}}{\mathbf{H}_{ik}} = 0. \end{aligned} \quad (50)$$

We have

$$\begin{aligned} &-2\mathbf{B}_{ik}^+(\mathbf{H}')_{ik}^2 + 2\mathbf{B}_{ik}^-\mathbf{H}_{ik}^2 + 2(\mathbf{H}'\mathbf{A}^+)_{ik}\mathbf{H}_{ik}^2 \\ &-2(\mathbf{H}'\mathbf{A}^-)_{ik}(\mathbf{H}')_{ik}^2 + 2\lambda(\mathbf{L}^+\mathbf{H}')_{ik}\mathbf{H}_{ik}^2 \\ &-2\lambda(\mathbf{L}^-\mathbf{H}')_{ik}(\mathbf{H}')_{ik}^2 = 0. \end{aligned} \quad (51)$$

Therefore

$$\mathbf{H}_{ik} = \mathbf{H}'_{ik} \sqrt{\frac{\mathbf{B}_{ik}^+ + (\mathbf{H}'\mathbf{A}^-)_{ik} + \lambda(\mathbf{L}^-\mathbf{H}')_{ik}}{\mathbf{B}_{ik}^- + (\mathbf{H}'\mathbf{A}^+)_{ik} + \lambda(\mathbf{L}^+\mathbf{H}')_{ik}}}. \quad (52)$$

The Hessian matrix containing the second derivatives

$$\frac{\partial^2 Z(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{ik} \partial \mathbf{H}_{jl}} = \delta_{ij} \delta_{kl} \mathbf{Y}_{ik} \quad (53)$$

is a diagonal matrix with positive entries

$$\begin{aligned} \mathbf{Y}_{ik} &= \frac{4[(\mathbf{B}^+)_{ik} + (\mathbf{H}'\mathbf{A}^-)_{ik} + \lambda(\mathbf{L}^-\mathbf{H}')_{ik}]\mathbf{H}'_{ik}}{\mathbf{H}_{ik}^2} \\ &+ 2\frac{\mathbf{B}_{ik}^- + (\mathbf{H}'\mathbf{A}^+)_{ik} + \lambda(\mathbf{L}^+\mathbf{H}')_{ik}}{\mathbf{H}'_{ik}}. \end{aligned} \quad (54)$$

Thus, $Z(\mathbf{H}, \mathbf{H}')$ is a convex function of \mathbf{H} . Therefore, we obtain the global minimum by setting $\partial Z(\mathbf{H}, \mathbf{H}')/\partial \mathbf{H}_{ik} = 0$ in (41) and solving for \mathbf{H} . Rearranging, we obtain (21). \square

Theorem 7. *Updating \mathbf{V} using (21) will monotonically decrease the value of the objective in (13); hence it converges.*

Proof. By Lemma 2 and Theorem 4, we can get that $\mathcal{F}(\mathbf{V}^t) = Z(\mathbf{V}^{(t)}, \mathbf{V}^{(t)}) \geq Z(\mathbf{V}^{(t+1)}, \mathbf{V}^{(t)}) \geq \mathcal{F}(\mathbf{V}^{(t+1)})$, so $\mathcal{F}(\mathbf{V})$ is monotonically decreasing. Since $\mathcal{F}(\mathbf{V})$ is obviously bounded below, we prove this theorem. \square

4. Experimental Results

In this section, we show the performance of the proposed method on face recognition and compare our proposed method with the popular subspace learning algorithms: four unsupervised ones which are principal component analysis [21] (PCA), neighborhood preserving embedding (NPE) [20], local nonnegative matrix factorization (LNMF) [4], and convex nonnegative factorization (CNMF) [13] the one supervised algorithm and which is linear discriminant analysis (LDA) [21]. We use the nearest neighbor (NN) classifier as baseline in original space. We apply different algorithms to obtain new representations for each chosen data set, and then the NN method is applied in the new representation spaces.

4.1. Data Preparation. The experiments are used on three data sets. One is Cambridge ORL face database, the other is the Yale database, and the third one is the CMU PIE face database. The important statistics of these data sets are described below.

The Yale database contains 165 gray scale images of 15 individuals. All images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses.

The ORL database contains ten different images of each of 40 distinct subjects, thus 400 images in total. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

TABLE 1: Face recognition accuracy on the ORL data set. The number in brackets is the corresponding projection dimensionality.

Method	2 Train	3 Train	4 Train
Baseline	69.32%	77.56%	83.48%
PCA	69.32% (79)	77.56% (118)	83.48% (152)
LDA	72.80% (25)	83.79% (39)	90.13% (39)
NPE	73.19% (36)	84.29% (54)	91.06% (73)
NMF	70.87% (97)	78.98% (81)	84.48% (95)
LNMF	71.73% (178)	81.09% (168)	86.31% (195)
CNMF	72.23% (138)	83.58% (143)	89.56% (111)
NPCNMF	77.31% (143)	86.73% (153)	93.35% (145)

The CMU PIE face database contains more than 40 000 facial images of 68 people. The images were acquired over different poses, under variable illumination conditions, and with different facial expressions. In our experiment, we choose the images from the frontal pose (C27) and each subject has around 49 images from varying illuminations and facial expressions.

In all the experiments, images are preprocessed so that faces are located. Original images are first normalized in scale and orientation such that the two eyes are aligned at the same position. Then the facial areas were cropped into the final images for clustering. Each image is of 32×32 pixels with 256 gray levels per pixel.

4.2. Parameter Settings. For each data set, we randomly divide it into training and testing sets, and evaluate the recognition accuracy on the testing set. In detail, for each individual in the ORL and Yale data sets, we randomly select 2, 3, and 4 images per individual, respectively, for training samples, and the remaining for test samples, while for each individual in the PIE data set, we randomly select 5, 10, and 20 images per individual for training samples. For each partition, we repeated each experiment 20 times and calculated the average recognition accuracy. In general, the recognition rate varies with the dimension of the face subspace. The best result obtained in the optimal subspace and the corresponding dimensionality for each method are shown.

For the face recognition experiments, several parameters need to be decided beforehand. For LDA, we use PCA as a first step dimensionality reduction algorithm to avoid the singularity problem. The dimension of the PCA step is fixed as $N - c$ and then performs LDA. There are two parameters in our NPCNMF and NPE approach: the number of nearest neighbors k and the regularization parameter λ . Throughout our experiments, we empirically set the number of nearest neighbors k to 5, the value of the regularization parameter λ to 100.

Each testing sample y is projected into the linear subspace spanned by the column vectors of the basis matrix U , namely, $\mathbf{h}_y \approx \mathbf{W}^\dagger y$, where \mathbf{W}^\dagger indicates the pseudoinverse of matrix \mathbf{W} .

4.3. Classification Results. Tables 1, 2, and 3 show the evaluation results of all the methods on the three data sets,

TABLE 2: Face recognition accuracy on the Yale data set. The number in brackets is the corresponding projection dimensionality.

Method	2 Train	3 Train	4 Train
Baseline	46.04%	49.96%	55.62%
PCA	46.04% (29)	49.96% (44)	55.62% (58)
LDA	42.81% (11)	60.33% (14)	68.10% (13)
NPE	48.19% (13)	62.00% (19)	69.00% (73)
NMF	44.11% (112)	49.00% (195)	52.19% (164)
LNMF	44.00% (157)	48.84% (198)	53.57% (197)
CNMF	49.72% (125)	59.50% (168)	65.77% (129)
NPCNMF	63.45% (124)	71.83% (148)	81.38% (153)

TABLE 3: Face recognition accuracy on the PIE data set. The number in brackets is the corresponding projection dimensionality.

Method	5 Train	10 Train	20 Train
Baseline	43.02%	62.90%	83.19%
PCA	42.87% (199)	62.51% (195)	82.84% (200)
LDA	84.39% (67)	90.47% (67)	93.98% (67)
NPE	84.71% (166)	91.48% (200)	94.33% (200)
NMF	78.66% (200)	88.98% (200)	92.52% (200)
LNMF	76.47% (200)	87.91% (200)	92.61% (196)
CNMF	83.72% (176)	90.89% (187)	93.78% (159)
NPCNMF	88.43% (147)	94.86% (158)	98.58% (133)

respectively, where the value in each entry represents the average recognition accuracy of 20 independent trials, and the number in brackets is the corresponding projection dimensionality. These experiments reveal a number of interesting points.

- (1) It is clear that the use of dimensionality reduction is beneficial in face recognition. There is a significant increase in performance from using LDA, NPE, NMF, LNMF, and CNMF. However, PCA fails to gain improvement over the baseline. This is because that PCA does not encode the discriminative information.
- (2) The performances of nonnegative algorithms NMF, LNMF, and CNMF are much worse than supervised algorithms LDA, which shows that without considering the labeled data, nonnegative algorithms could not guarantee good discriminating power.
- (3) Our NPCNMF algorithm outperforms all other five methods. The reason lies in the fact that NPCNMF considers the geometrical structure of the data and achieves better performance than the other algorithms. This shows that by leveraging the power of both the parts-based representation and the intrinsic geometrical structure of the data, NPCNMF can learn a better compact representation in the sense of semantic structure.

5. Conclusion and Future Work

In this paper, we have presented a novel matrix factorization method called NPCNMF for dimensionality reduction,

which respects the local geometric structure. As a result, NPCNMF can discriminate power more than the ordinary NMF and CNMF approaches which only consider the Euclidean structure of the data. Experimental results on face datasets show that NPCNMF provides better representation in the sense of semantic structure.

Several challenges remain to be investigated in our future work.

- (1) A suitable value of λ is important to our algorithm. It remains unknown how to do model selection theoretically and efficiently.
- (2) NPCNMF is currently limited to the linear projections, and those nonlinear techniques (e.g., kernel tricks) may further boost the algorithmic performance. We will investigate it in our future work.
- (3) Another further research direction is how to extend the current framework for tensor-based nonnegative data decomposition.
- (4) NPCNMF algorithm is iterative and sensitive to the initialization of \mathbf{W} and \mathbf{H} . It is unclear how to choose optimal initialization parameters in a principled manner.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [2] D. D. Lee and H. S. Seung, "Algorithms for nonnegative matrix factorization," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS '00)*, 2000.
- [3] M. Cooper and J. Foote, "Summarizing video using nonnegative similarity matrix factorization," in *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, 2002.
- [4] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CPRV '01)*, pp. I207–I212, December 2001.
- [5] P. Paatero and U. Tapper, "Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.
- [6] P. O. Hoyer, "Non-negative sparse coding," in *Processings of IEEE Workshop on Neural Networks for Signal Processing*, 2002.
- [7] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2003.
- [8] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3D non-negative tensor factorization," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 50–57, October 2005.
- [9] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 793–800, August 2005.
- [10] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Fisher non-negative matrix factorization for learning local features," in *Proceedings of the 6th Asian Conference on Computer Vision (ACCV '04)*, 2004.
- [11] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 683–695, 2006.
- [12] I. Kotsia, S. Zafeiriou, and I. Pitas, "Novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 588–595, 2007.
- [13] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.
- [14] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [15] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [16] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.
- [17] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 902–913, 2011.
- [18] L. Zhang, L. Qiao, and S. Chen, "Graph-optimized locality preserving projections," *Pattern Recognition*, vol. 43, no. 6, pp. 1993–2002, 2010.
- [19] X. He and P. Niyogi, "Locality preserving projections," in *Proceedings of the NIPS*, 2003.
- [20] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proceedings 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 1208–1213, October 2005.
- [21] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

Research Article

Moving Object Detection and Shadow Removing under Changing Illumination Condition

Jinhai Xiang,^{1,2} Heng Fan,² Honghong Liao,¹ Jun Xu,³ Weiping Sun,¹ and Shengsheng Yu¹

¹ School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

² College of Science, Huazhong Agricultural University, Wuhan 430070, China

³ Department of Physics, Center China Normal University, Wuhan 430079, China

Correspondence should be addressed to Shengsheng Yu; ssyu@mail.hust.edu.cn

Received 5 October 2013; Accepted 28 December 2013; Published 24 February 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Jinhai Xiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Moving object detection is a fundamental step in video surveillance system. To eliminate the influence of illumination change and shadow associated with the moving objects, we proposed a local intensity ratio model (LIRM) which is robust to illumination change. Based on the analysis of the illumination and shadow model, we discussed the distribution of local intensity ratio. And the moving objects are segmented without shadow using normalized local intensity ratio via Gaussian mixture model (GMM). Then erosion is used to get the moving objects contours and erase the scatter shadow patches and noises. After that, we get the enhanced moving objects contours by a new contour enhancement method, in which foreground ratio and spatial relation are considered. At last, a new method is used to fill foreground with holes. Experimental results demonstrate that the proposed approach can get moving objects without cast shadow and shows excellent performance under various illumination change conditions.

1. Introduction

Moving object detection is a fundamental step in many image analysis applications including automated visual surveillance, video indexing, and human machine interaction. Many researches on moving object detection have been proposed, such as background subtraction, optical flow, and temporal differencing [1]. However, moving object detection techniques are often affected by factors such as shadow, illumination changes, and noise. Generally, different factors cause different consequences. In this paper, we focus on shadow and illumination factors.

The shadow causes many problems in object localization, segmentation, object detection, and tracking. Furthermore, the shadow may cause the following problems: objects merge with each other; object shapes may be altered; the background may be misclassified as foreground, and objects are missed [2]. Shadows associated with moving objects can easily be misinterpreted as additional objects. At the same time,

if illumination changes, some background pixels may be detected as foreground pixels, which makes it hard to obtain the clean moving objects. Therefore, eliminating shadows and handling illumination changes have a great effect on the performance of subsequent steps such as tracking, recognition, classification, and activity analysis, which need the accurate detection of a moving object and the acquisition of its exact shape.

Some reviews about shadow detection have been reported in the literature [2–4], from the views of physical and geometrical to heuristic techniques. Most of the real-time shadow detection techniques work at pixel level and use color information for shadow detection directly or indirectly, wholly or partially. Shadow detection methods can be roughly divided into two categories: based on statistic and based on the video features.

The principle of statistic-based methods is to build pixel-based statistical models of detecting cast shadows. In [5], Zivkovic and van der Heijden use Gaussian mixture model

(GMM) to detect moving cast shadows. This method consists of building a GMM for moving objects, identifying the distribution of moving objects and shadows and modifying the learning rates of the distributions. Joshi and Papanikolopoulos [6] propose semisupervised learning technique to tackle the static setting with human input for detecting shadows. Jung [7] uses a statistical approach combined with geometrical constraints for detecting shadows, but it is only for grayscale video sequences. The statistic-based methods identify distribution of shadow pixel value and are robust in different scenes. Although these methods reduce false hits of property descriptions (i.e., color or intensity) of shadow, it cannot eliminate them. Generally, these methods require training video sequences, and shadow must be extracted from training video sequences by hand. Furthermore, it is hard to operate online and in real time because these methods need an additional learning step.

The video features-based methods are based on the fact that video features, including geometric, color, gradient, and brightness, and so forth, are different in shadow, background, and moving objects. These methods are more general than those by statistic. The methods based on image features can be further divided into the following four categories [2]: chromaticity-based, light physical characteristics-based, geometric relations-based, and texture-based.

The chromaticity-based methods are mostly on the basis of a single pixel and combined with color information to test the shadow. These methods make certain assumptions about the shadow properties: (1) a shadow darkens the background area on which it falls; (2) a shadow falls on the background plane; (3) a shadow changes luminance of an area more significantly than color [8]. If these assumptions are not satisfied, the accuracy of color-based approaches for shadow detection will decrease obviously. For better separation between intensity and chromaticity, several color spaces such as HSV [9], $c1c2c3$ [10], HSL [11], and RGB [12] have been developed to detect moving cast shadow robustly. Most of these methods are computationally inexpensive and easy to implement. However, they are sensitive to noise and will fail when shadow regions are darker or moving objects have similar color information with background.

The light physical characteristics-based methods are based on the linear attenuation model of the light intensity which assumes that illumination source produces pure white light. In outdoor environment, the main light source is sunlight (white) and the reflected light comes from the sky (blue). Generally, other light source is influenced by sunlight. If sunlight is blocked, the effect of reflected light from the sky increases, and the chromaticity of shaded region is shifted toward the blue component. Nadimi and Bhanu [13] presented a dichromatic model which takes into account two light sources to predict the color changes of the shaded region effectively. Further works consider a variety of different light intensity conditions and build a more general nonlinear attenuation model to adapt to the indoor and outdoor scenes [14, 15]. These methods are still using the chromaticity characteristics, and if the foreground color is closer to the

cast shadow, there would be some mistakes. The geometric feature-based methods mainly consider that, under certain light source, shape, and size of shadow, position relationship between objects can be ensured and segment shadow from foreground [16–18]. These methods do not need background model; however, they need detailed shape information and other information of foreground targets. These methods are limited to detect specific objects, and furthermore they need position relationship between objects and shadow. In the case of multiple shadows and multiple objects, these methods do not work well.

The texture-based methods assume that the texture of shaded region is invariant. These methods generally include the following steps: (1) detect foreground with shadow and (2) classify foreground as either foreground or shadow based on the texture correlation. If texture of candidate area is similar to that of background, it may be misclassified as the shadow. Leone and Distante [19] propose a moving cast shadows method based on Gabor functions and matching pursuit strategy. Zhang et al. [20] employ ratio edge as the ratio between the intensity of one pixel and its neighboring pixels to detect shadows. Confirming the existence of shadows, Xiao et al. [21] reconstruct coarse object shapes and then extract cast shadows by subtracting moving objects from one changed mask. The texture-based methods are effective without colors information and robust to illumination changes. However, texture-based shadow detection methods need to compare adjacent pixels, and their complexity is high.

Generally, we must obtain true foreground pixels before shadow detection by using methods above. A background subtraction method such as Gaussian mixture model (GMM) and its modified versions are some representative methods to detect moving pixels with shadow. However, in a dynamic scene, the varying background is detected as a moving object. In order to remove shadow directly and make it robust to illumination variations, we present a local intensity ratio model (LIRM), which shows illumination invariance. First, we use normalized local intensity ratio to replace the pixel to detect moving object without shadow via Gaussian mixture model. Second, erosion is used to get the moving objects contours and erase the scatter shadow patches and noises. After that, we get the enhanced moving object contours by contour enhancement method from erosion image and foreground image. At last, we use the local foreground density, and contour orientation to fill enhanced moving objects with holes. Experimental results demonstrate that the proposed approach can get moving objects without cast shadow and shows excellent performance under various changing illumination conditions.

This paper is organized as follows. Section 2 introduces illumination change model, presents LIRM, and analyzes its distribution. The process of Gaussian mixture model for foreground detection and corresponding postprocess algorithm are described in Section 3. Section 4 analyzes foreground detection results in four test videos with different light conditions and compares the results with other methods. Finally Section 5 concludes the paper.

2. Illumination Change Model

2.1. Local Intensity Ratio and Illumination Invariant. Zhang et al. [20] proved that the ratio edge is illumination invariant and use it to classify each moving pixel into foreground object or moving shadow. Inspired by this work, we define the local intensity ratio and analyze its illumination invariance.

First, the definition of the local intensity ratio (LIR) is

$$R(x, y) = \frac{p(x, y) \cdot |A(x, y)|}{\sum_{(i, j) \in A(x, y)} p(i, j)}, \quad (1)$$

where $p(x, y)$ is the intensity of pixel (x, y) ; $|A(x, y)|$ is the number of pixels in a local region; $A(x, y)$ is the local region of given pixel (x, y) (a rectangle region) and defined as follows:

$$A(i, j) = \{(x + i, y + j) \mid -r \leq i, j \leq r\} \quad (2)$$

and r is the size of the local region.

For the images or videos acquired by a fixed camera, the intensity value of the pixel is

$$p(x, y) = \rho(x, y) \cdot I(x, y) \cdot \theta(x, y), \quad (3)$$

where $\rho(x, y)$ is the reflectance of object surface in pixel (x, y) , that is, the reflection coefficient; $I(x, y)$ is the amount of light power per receiving object surface area in pixel (x, y) ; $\theta(x, y)$ is the sensor sensitivity of the camera. The light in scenes can be divided into direct light of light source and scattered light of environment. We assume that the light source is a distant light source, and the light is parallel light, such as sunlight. Light of environment is scattered light, light direction is random, and its intensity in scenes is assumed to be constant. If objects are occluded, the scene can be divided into three cases: illuminated area, penumbra area, and umbra area [22, 23], and the light intensity of the real target pixel (x, y) is

$$I(x, y) = \begin{cases} L_a + L_p \cdot \cos \phi(x, y), & \text{illuminated area,} \\ L_a + \alpha(x, y) \cdot L_p \\ \quad \cdot \cos \phi(x, y), & \text{penumbra area,} \\ L_a, & \text{umbra area,} \end{cases} \quad (4)$$

where L_a is the intensity of ambient light; L_p is the intensity of light source; $\alpha(x, y)$ is the transition inside the penumbra which depends on the light source and scene geometry, and $0 \leq \alpha(x, y) \leq 1$; $\phi(x, y)$ is the angle between light source direction and surface normal.

While using RGB model to express light intensity, (4) can be shown as follows

$$I(x, y)^k = \begin{cases} L_a^k + L_p^k \cdot \cos \phi(x, y), & \text{illuminated area,} \\ L_a^k + \alpha(x, y) \cdot L_p^k \\ \quad \cdot \cos \phi(x, y), & \text{penumbra area,} \\ L_a^k, & \text{umbra area,} \end{cases} \quad (5)$$

where k is the number of color channels, $k = 1, 2, \text{ and } 3$. For simplification, in this paper we consider only one light source and ignore the influence of different color channels.

The analysis of the local intensity ratio under the three different scenes is as shown in Figure 1.

If all the local regions belong to one of the three areas shown in Figure 1, that is, $A(x, y) \in A1$ or $A(x, y) \in A2$, or $A(x, y) \in A3$, and $A1, A2$, and $A3$ express the illuminate area, penumbra area, and umbra area, respectively, we can obtain the following results by formulas (4), (3), and (2) and assume $|A(x, y)| = n$.

Case 1 ($A(x, y) \in A1$). Consider

$$R_1(x, y) = \frac{n(L_a + L_p \cdot \phi(x, y)) \cdot \rho(x, y) \cdot \theta(x, y)}{\sum_{(i, j) \in A(x, y)} (L_a + L_p \cdot \phi(i, j)) \cdot \rho(i, j) \cdot \theta(i, j)}. \quad (6)$$

Case 2 ($A(x, y) \in A2$). Consider

$$R_2(x, y) = \frac{n(L_a + \alpha(x, y) \cdot L_p \cdot \phi(x, y)) \cdot \rho(x, y) \cdot \theta(x, y)}{\sum_{(i, j) \in A(x, y)} (L_a + \alpha(i, j) \cdot L_p \cdot \phi(i, j)) \cdot \rho(i, j) \cdot \theta(i, j)}. \quad (7)$$

Case 3 ($A(x, y) \in A3$). Consider

$$R_3 = \frac{n \cdot \rho(x, y) \cdot \theta(x, y)}{\sum_{(i, j) \in A(x, y)} \rho(i, j) \cdot \theta(i, j)}. \quad (8)$$

If the pixels are in the same local region, we assume that (1) pixels in the same local region belong to the same object plane, thus $\phi(x, y) = \phi(i, j)$; (2) sensor sensitivity of camera in all local regions is the same; $\theta(x, y) = \theta(i, j)$; (3) intensity of penumbra is also the same; $\alpha(x, y) = \alpha(i, j)$.

According to the previous assumptions, we get

$$\begin{aligned} \theta(x, y) &= \theta(i, j), \\ \cos \phi(x, y) &= \cos \phi(i, j), \\ \alpha(x, y) &= \alpha(i, j), \end{aligned} \quad (9)$$

where $(i, j) \in A(x, y)$.

By the previous assumptions, and formulas (6), (7), (8), and (9), we also get

$$R_1(x, y) = \frac{n \cdot \rho(x, y) \cdot \theta(x, y)}{\sum_{(i, j) \in A(x, y)} \rho(i, j) \cdot \theta(i, j)} = \text{const}, \quad (10)$$

$$R_2(x, y) = \frac{n \cdot \rho(x, y) \cdot \theta(x, y)}{\sum_{(i, j) \in A(x, y)} \rho(i, j) \cdot \theta(i, j)} = \text{const}, \quad (11)$$

$$R_1(x, y) = R_2(x, y) = R_3(x, y) = \text{const}. \quad (12)$$

If local region belongs to one of the three illumination cases, local intensity ratio is influenced only by reflectivity. Therefore, local intensity ratio is related to the reflectivity of target surface, not to the illumination changes and types, that is, illumination invariance.

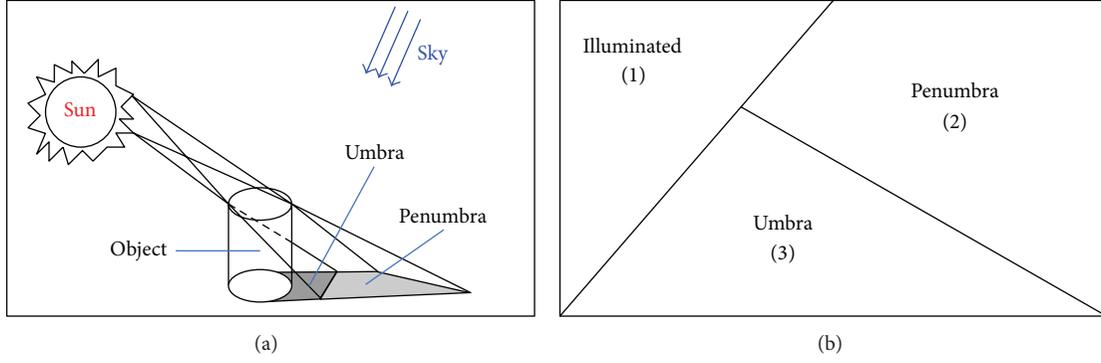


FIGURE 1: The illumination of the scene (three cases).

The local intensity ratio of pixels is constant under different illumination conditions according to formula (12). As a result, local intensity ratio model not only removes the influence of illumination, but also eliminates shadow of the foreground targets.

2.2. Distribution of Local Intensity Ratio. Generally, it is assumed that the images are corrupted by Gaussian white noise, which can be expressed as

$$P_r(x, y) = P_b(x, y) + \varepsilon(x, y), \quad \varepsilon(x, y) \sim N(0, \sigma^2), \quad (13)$$

where $P_r(x, y)$ represents actual pixel value; $P_b(x, y)$ denotes real pixel value in scene; $\varepsilon(x, y)$ represents noise. The actual local intensity ratio is defined as

$$\begin{aligned} R_r &= \frac{P_r(x, y)}{\sum_{(i,j) \in A(x,y)} P_r(i, j)} \\ &= \frac{P_b(x, y) + \varepsilon(x, y)}{\sum_{(i,j) \in A(x,y)} (P_b(i, j) + \varepsilon(i, j))}. \end{aligned} \quad (14)$$

Usually, $P_b(i, j) \gg \varepsilon(i, j)$; formula (14) can be denoted as

$$\begin{aligned} R_r &= \frac{P_b(x, y)}{\sum_{(i,j) \in A(x,y)} P_b(i, j)} \\ &\quad + \frac{\varepsilon(x, y)}{\sum_{(i,j) \in A(x,y)} P_b(i, j)} \\ &= R_b + \frac{\varepsilon(x, y)}{\sum_{(i,j) \in A(x,y)} P_b(i, j)}, \end{aligned} \quad (15)$$

where R_b is the real local intensity ratio:

$$R_b = \frac{P_b(x, y)}{\sum_{(i,j) \in A(x,y)} P_b(i, j)}. \quad (16)$$

If pixel value in scene is constant, $P_b(x, y) / \sum_{(i,j) \in A(x,y)} P_b(i, j)$ obeys Gaussian white noise distribution. Therefore, local intensity ratio and pixel value have the same

distribution. On the other hand, if $\sum_{(i,j) \in A(x,y)} P_b(i, j)$ is small, the white noise will be amplified.

Theoretically, the local intensity ratio ranges from 0 to infinite. In order to be consistent with the scope of a pixel value, the normalized definition of the local intensity ratio is as follows:

$$\text{LR}(x, y) = \left(\frac{2}{1 + e^{-R(x,y)}} - 1 \right) \times 255. \quad (17)$$

3. Moving Object Detection Based on Local Intensity Ratio

In this section, we present the detailed procedure of shadow removing. The proposed method is a multistage approach and the flow chart is shown in Figure 2.

Generally, the shadow detection methods utilize object detection or image segmentation algorithms to detect the foreground with shadow and then classify the foreground as foreground or shadow. According to the results in the previous section, the local intensity ratio obeys Gaussian distribution. This paper uses the normalized LIR to replace the pixel to detect foreground, and the process is shown in Figure 2. In this paper, Gaussian mixture model [5] is used to acquire the foreground. In mixture of Gaussians, the recent history of each pixel is maintained using k Gaussian distributions and the value of k typically chooses from 3 to 5. Each distribution has its associated attributes like weight ω_k , mean μ_k , and variance σ_k . Each Gaussian updates its parameters using LIR in every new frame.

After detecting foreground, the foreground target with little shadow can be obtained (called foreground image as follows, I_{fore}). Meanwhile, some parts of the foreground object may be darker than the background and easily be false detected as background. The false detected parts of shadow are discrete; the false detected parts of moving object like holes are surrounded by foreground pixels (such as contour). The moving objects contour can be detected, while the shadows contour can be removed at most time. In order to get enhanced moving objects without discrete shadow patches, erosion is used to erase the scatter and noises and to get moving objects contours image without shadow (called contour image as follows, I_{cont}). Then the contour image is

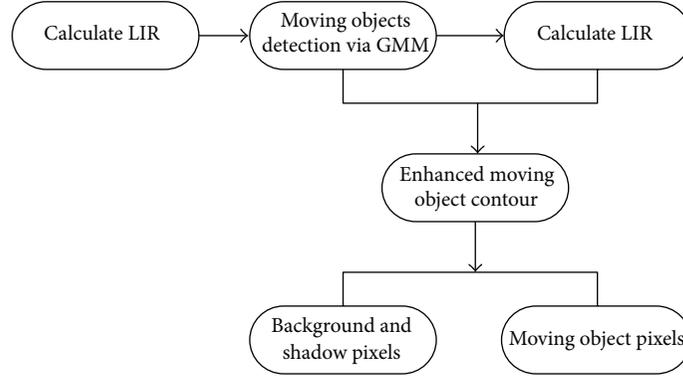


FIGURE 2: Flow chart of moving object detection and shadow removing.

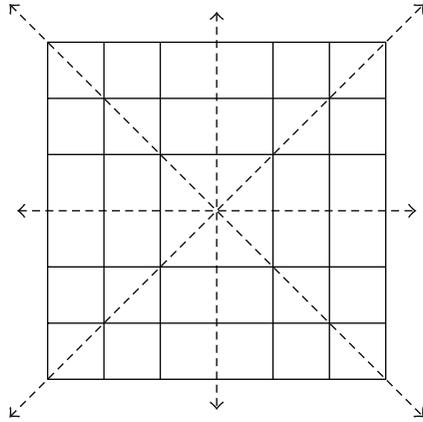


FIGURE 3: Local region of the center pixel detection and its eight directions.

used to enhance moving objects by contour enhancement method. The enhancement procedure is as follows: for a pixel $P(x, y)$ being not foreground pixel in contours image, we calculate the weighted foreground pixel ratio of a local region in the contour image and foreground image as follows:

$$s(x, y) = \frac{(\omega_c N_c(x, y) + \omega_f N_f(x, y))}{|A(x, y)|}, \quad (18)$$

$$N_c(x, y) = |\{(i, j) \mid (i, j) \in A(x, y) \wedge (i, j) \in I_{\text{cont}}\}|,$$

$$N_f(x, y) = |\{(i, j) \mid (i, j) \in A(x, y) \wedge (i, j) \in I_{\text{fore}}\}|,$$

where $s(x, y)$ represents the rate of the enhanced moving object contours in (x, y) , $N_c(x, y)$ is the number of foreground pixels in a neighbor region of pixel (x, y) in contour image, $N_f(x, y)$ is the number of foreground pixels in a neighbor region of pixel (x, y) in foreground image, and ω_c and ω_f are the weight to contour image and foreground image. We select $\omega_c = 3$, $\omega_f = 1$ in our experiments. $|A(x, y)|$ is the number of pixels in a local region; $A(x, y)$ is the local region of given pixel (x, y) (a rectangle region) and defined as in (2).

On the other hand, we consider eight directions of a pixel, determine whether there is foreground pixel in each direction

of the local region and count how many times it reaches foreground pixels, as shown in Figure 3:

$$\delta_i(x, y) = \begin{cases} 1, & \text{foreground pixels in the } i\text{th direction,} \\ 0, & \text{otherwise,} \end{cases}$$

$$v(x, y) = \frac{1}{8} \sum_{i=1}^8 \delta_i(x, y), \quad (19)$$

where δ_i means whether there is a foreground pixel in the i th direction, and if it exists, its value is 1, otherwise 0. $v(x, y)$ is the ratio of foreground in eight directions. After calculating $s(x, y)$ and $v(x, y)$, by verifying whether or not the $s(x, y)$ and $v(x, y)$ of a pixel is in the feasible range, the pixel is determined by

$$p(x, y) = \begin{cases} 255, & v(x, y) > \gamma_v, s(x, y) > \gamma_s, \\ 0, & \text{otherwise,} \end{cases} \quad (20)$$

where γ_v and γ_s are the threshold values.

So, we get the enhanced foreground contours image (called enhanced foreground contour image, I_{enfc}). However, some moving object pixels are false detected as background pixels, such as holes in the image. To get integral foreground object, we use filling method to eliminate false detected pixel for background pixel (Figure 4). For a pixel (x, y) which is detected as background pixel in enhanced foreground contour image, we first consider eight directions of a pixel: determine whether there is foreground pixel in each direction of the local region and count how many times it reaches foreground pixels, as expressed in (19). We also consider the ratio in a local region between numbers of foreground pixels in foreground image and in contour image and the numbers of pixels in the same region:

$$s'(x, y) = \frac{N_s(x, y)}{N_r(x, y)}, \quad (21)$$

where $s'(x, y)$ is the ratio of pixels detected as foreground in local region, $N_s(x, y)$ is the number of pixels which are detected as foreground, and $N_r(x, y)$ is the sum of pixels in the local region.

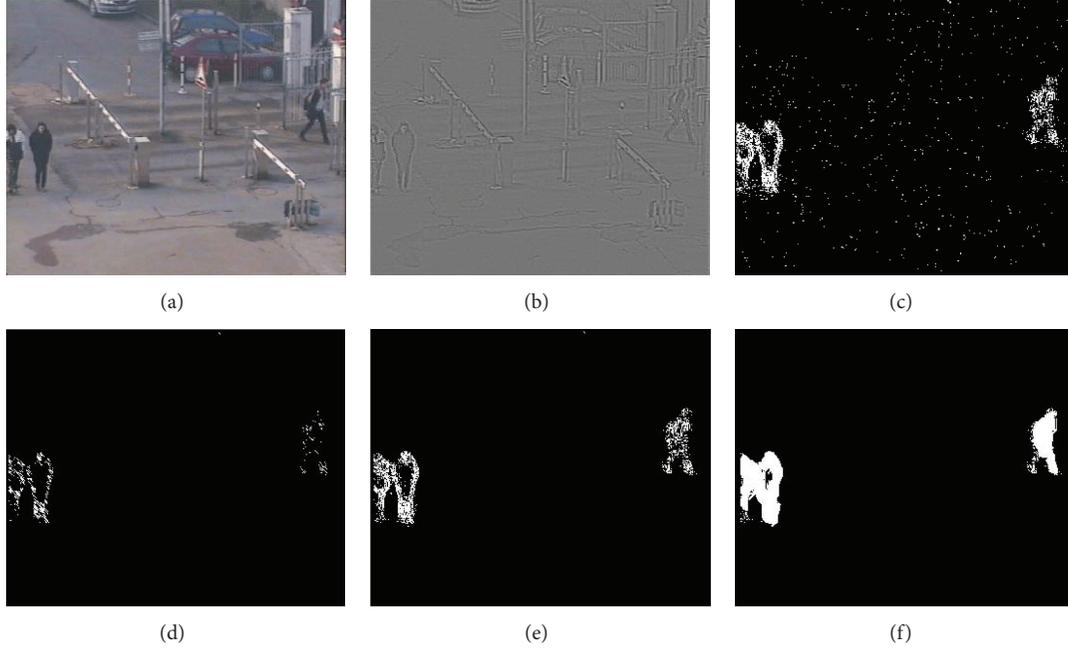


FIGURE 4: Illustration of the shadow removal process using Campus sequence. (a) Current frame. (b) Normalized LIR of current frame. (c) Foreground object detected using GMM of LIR. (d) Result of erosion operation (contour image) to remove false detection shadow and noise. (e) Enhanced contour image by contour enhancement method. (f) Final result of filling holes.

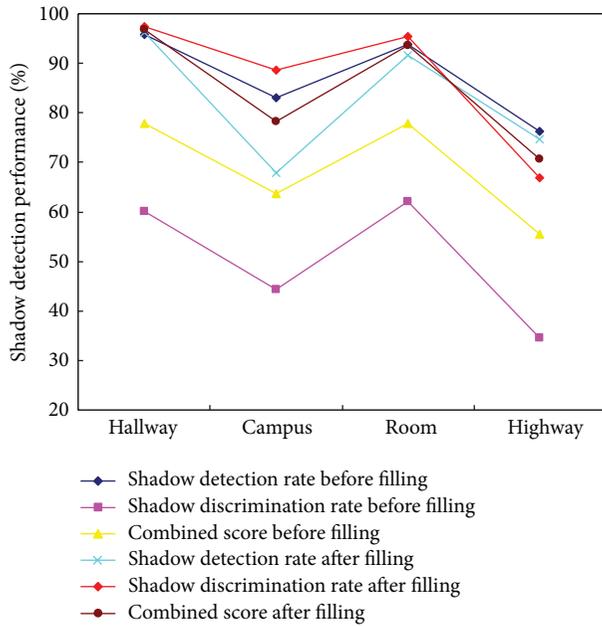


FIGURE 5: Compared detection result of LIRM via GMM and postprocess.

After calculating $v(x, y)$ and $s'(x, y)$, we can get the option that the pixel $P(x, y)$ is moving object pixel or not by

$$p(x, y) = \begin{cases} 255, & v(x, y) < \gamma'_v, s'(x, y) < \gamma'_s, \\ 0, & \text{otherwise,} \end{cases} \quad (22)$$

where γ'_v and γ'_s are the threshold values; $p(x, y)$ denotes the final value of pixel (x, y) .

4. Experiment Results

In this section, we describe the test videos used in this study, measure the performance of the proposed method, and compare the performance to the state-of-the-art methods. In general, state-of-the-art methods use different steps to remove shadow and reduce the influence of illumination changes. To compare with other methods in different steps, we first compare the shadow removing results and then compare the results under various illumination change conditions in different data sets.

4.1. Experiment Conditions and Parameter Setting. The size of the local rectangle region in (2) is important to the proposed method. If the size is too large, its inner region cannot meet the assumptions (9). On the other hand, if too small, the change of its value is also small. In our experiment, r is set as 2.

4.2. Experiment Results. To test the effect of shadow removing based on local intensity ratio and the effect of foreground detection, we use four typical videos from the ATON video set [4] as the test videos, which are Campus, Hallway, Highway 1, and Intelligent room. Intelligent room and Hallway are typical indoor environment. In the videos, people are walking in scene, and their shadows are mapped to the background. The Campus is a video of campus parking lot, and there are shadows of people and cars in the scenes.

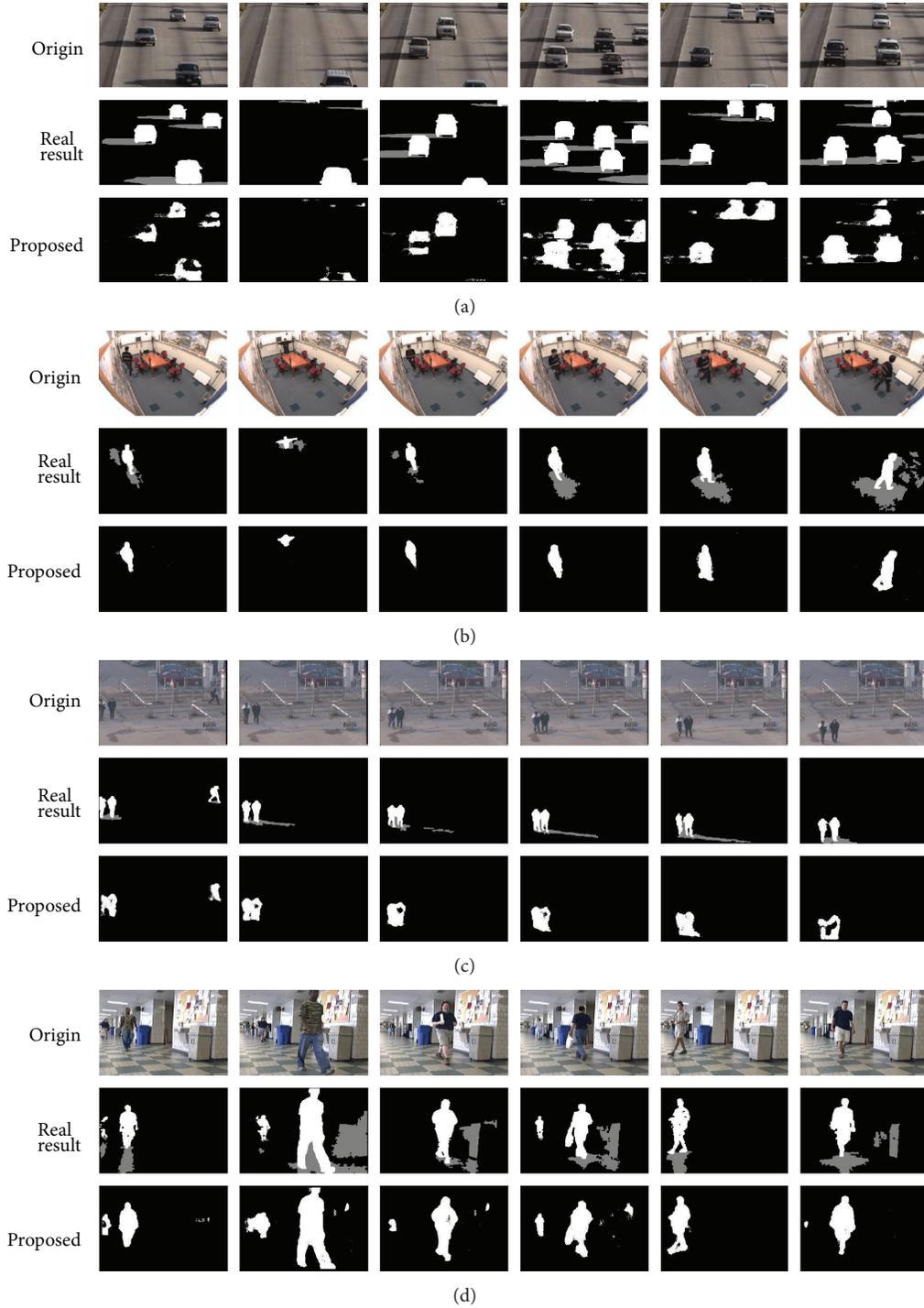


FIGURE 6: Detection results in different scenes. (a) is Highway I sequence, (b) is intelligent room sequence, (c) is Campus sequence, and (d) is Hallway sequence. In the image of detection result, the moving object pixels are white, the background is black, and the shadow is gray.

To compare the performance of the proposed moving objects detection method with other shadow detection methods, we assume that if shadow is detected as background, it is correct detection, otherwise it is false detection, and if moving object is detected as foreground, it is correct

detection, otherwise it is false detection. The proposed method is compared with five methods introduced by [2], which are chromaticity-based method, physical method, geometry-based method, Small region (SR) texture-based method, and large region (LR) texture-based method. To

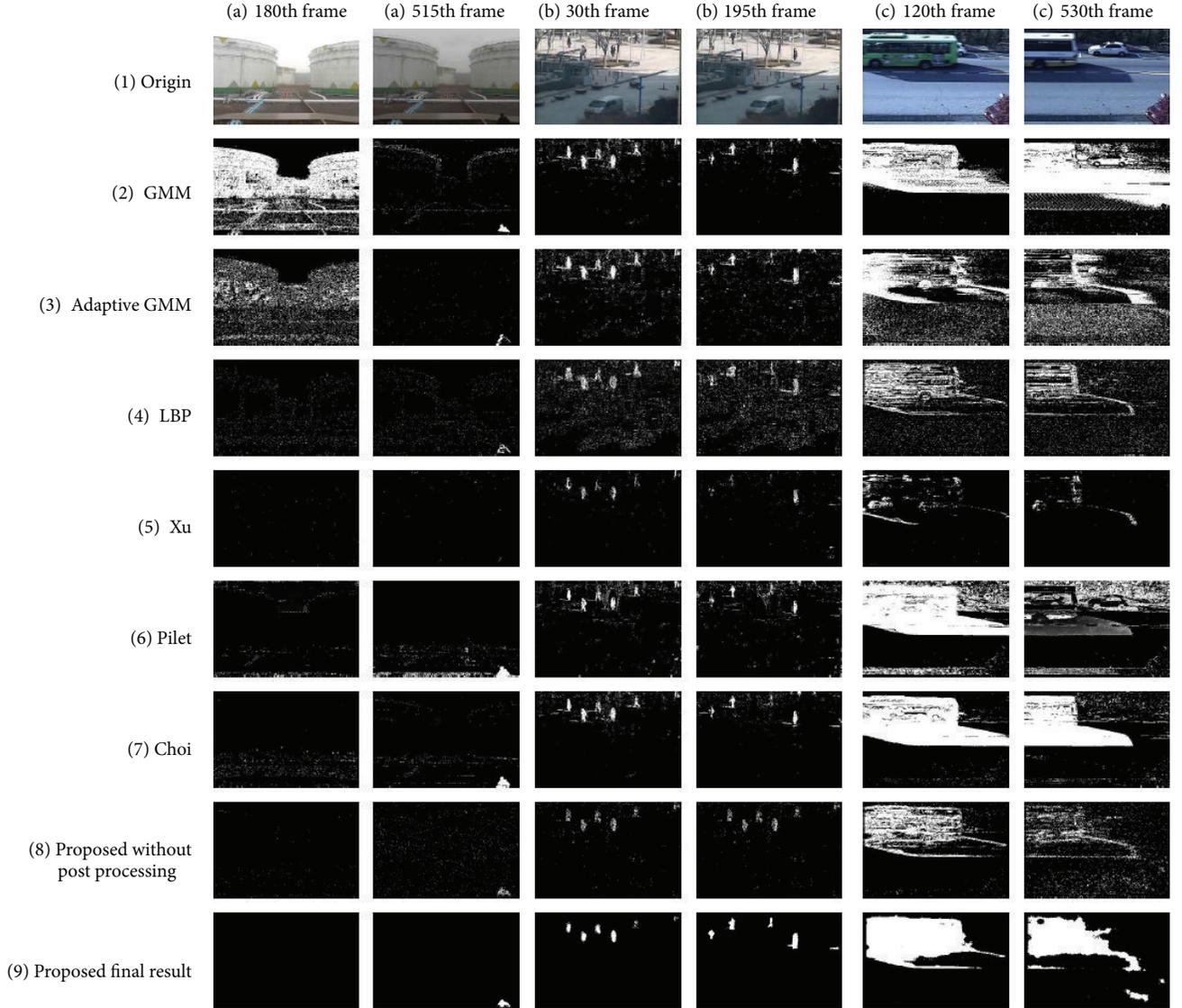


FIGURE 7: The comparison of moving object detection results under the changing illumination condition (three scenes). From left to right: (a) is video 1, (b) is video 2, and (c) is video 3. The 1st row is an input image, the remaining nine rows are detection results of each method, the 2nd row is a detection result of GMM, the 3rd row is of adaptive GMM, the 4th row is of LBP, the 5th row is of Xu, the 6th row is of Pilet, the 7th row is of Choi, the 8th row is of the proposed method without postprocessing, and the 9th row is the final result of proposed method.

estimate the effect of these methods, shadow detection rate (η) and shadow distinguishing rate (ξ) are the discriminate parameters [4], and their definitions are as follows:

$$\eta = \frac{TP_S}{TP_S + FN_S} \times 100\%, \quad \xi = \frac{TP_F}{TP_F + FN_F} \times 100\%, \quad (23)$$

where S is shadow; F is foreground. TP_S is the number of the pixels which are detected correctly; FN_S is the number of shadow pixels which are mistakenly detected as foreground; TP_F is the number of foreground pixels detected correctly after removing shadows. FN_F is the number of foreground pixels which are mistakenly detected as shadow. The shadow distinguishing rate is concerned with maintaining the pixels

which belong to the moving object as foreground. In this paper, we use the average of the two rates as a single performance measure (avg).

Table 1 is the compared results among the method based on LIRM and other methods in [2]. Table 1 shows the average shadow detection and discrimination rates on each test sequence. From the compared results, the method proposed in this paper is better at most time. The main cause is that the presented method directly detects the foreground from video data, but other methods extract the shadow from foreground by true background and foreground with shadow. Our method use local intensity ratio to replace the actual pixel values; complexity is lower. Figure 6 is the result of proposed method in moving object detection without shadow in different scenes.

TABLE 1: Comparison of moving object detection with different methods.

Methods	Campus			Intelligent room			Hallway			Highway I		
	η (%)	ξ (%)	Avg	η (%)	ξ (%)	Avg	η (%)	ξ (%)	Avg	η (%)	ξ (%)	Avg
chrMask	74.84	38.59	56.72	95.84	67.56	81.70	95.81	82.44	89.14	77.65	68.38	73.01
geoMask	76.78	85.96	81.36	57.35	73.87	65.61	50.68	77.81	64.24	65.34	74.15	69.74
lrTexMask	41.31	99.18	70.24	81.97	96.75	89.36	97.13	97.01	97.07	60.21	93.94	77.07
phyMask	71.41	30.60	51.00	77.96	85.95	81.95	85.82	86.89	86.35	48.15	79.47	63.81
srTexMask	64.6	91.13	77.86	93.12	56.64	74.88	98.40	67.15	82.77	14.79	94.87	54.82
proposed	67.97	88.68	78.32	91.70	95.36	93.53	96.26	97.47	96.86	74.59	66.82	70.71

Figure 5 represents compared results of the LIRM moving object detection via GMM and postprocess. From Figure 5, the postprocess obviously improves the shadow discrimination rate and slightly reduces the shadow detection rate. However, the average of the two rates is increased obviously. So, the postprocess enhances the moving object detection result.

Finally, we show that the proposed method rapidly adapts to variations in environment. To test moving object detection under illumination changes condition, we compare the result of other techniques which are GMM [5], adaptive GMM [24] that controls the learning rate of the GMM adaptively, LBP [25] that detects moving objects using texture and edges, Xu's methods [26] that use a color chromaticity, Pilet's methods [27] that use illumination and spatial likelihood, and Choi's methods [28] that develop chromaticity difference model and brightness model that estimates the intensity difference and intensity ratio of false foreground pixels. For the quantitative comparison, we have made three test videos [28]. Figure 7 represents moving object detection results of each method. From Figure 7, we find the postprocessing of the proposed method can strengthen the moving objects and remove the shadow patches and the noise, but it sometimes enhances the edge of part shadows incorrectly and makes moving objects' edge fuzzy. In a word, the proposed method adapts to the changed background completely and detects moving object pixels without shadow pixels successfully. Although some other methods are robust to illumination, their detection moving pixels results show that they cannot detect moving object pixels meanwhile removing shadow.

5. Conclusions

In this paper, we propose a method to detect moving object meanwhile removing cast shadow, which is robust in changing illumination condition. This paper presents a local intensity ratio model according to illumination model and proves its illumination invariance. Meanwhile, if the noise in video obeys the Gaussian distribution, the local intensity ratio also obeys the Gaussian distribution. In the process of Gaussian mixture model to obtain foreground, this paper replaces actual pixel value by local intensity ratio. Finally, postprocess methods are used to get pure moving object pixels without noise, such as erosion method to remove shadow pitch and noise which is false detected as foreground, foreground contour enhanced method to strengthen moving

object contour, and filling methods to deal with holes in foreground object. Experiment results demonstrate that the method presented by this paper can eliminate shadow on foreground effectively, and is robust in changing illumination condition. However, in some scenes that foreground is similar to background or foreground is similar to shadow, foreground may be easily detected as background. If illumination changes severely, such as turn on/off light in indoor scene, the background may be easily detected as foreground, and the performance drops.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Jinhai Xiang is supported by the Fundamental Research Funds for the Central Universities under Grant no. 2013QC024. Jun Xu is supported by the National Natural Science Foundation of China under Grant no. 11204099 and self-determined research funds of CCNU from the colleges basic research and operation of MOE. Weiping Sun is supported by the National Natural Science Foundation of China under Grant no. 61300140.

References

- [1] M. Piccardi, "Background subtraction techniques: a review," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 3099–3104, October 2004.
- [2] A. Sanin, C. Sanderson, and B. C. Lovell, "Shadow detection: a survey and comparative evaluation of recent methods," *Pattern Recognition*, vol. 45, no. 4, pp. 1684–1695, 2012.
- [3] N. Al-Najdawi, H. E. Bez, J. Singhai, and E. A. Edirisinghe, "A survey of cast shadow detection algorithms," *Pattern Recognition Letters*, vol. 33, no. 6, pp. 752–764, 2012.
- [4] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 918–923, 2003.
- [5] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006.

- [6] A. J. Joshi and N. P. Papanikolopoulos, "Learning to detect moving shadows in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2055–2063, 2008.
- [7] C. R. Jung, "Efficient background subtraction and shadow removal for monochromatic video sequences," *IEEE Transactions on Multimedia*, vol. 11, no. 3, pp. 571–577, 2009.
- [8] M. Shoaib, R. Dragon, and J. Ostermann, "Shadow detection for moving humans using gradient-based background subtraction," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 773–776, April 2009.
- [9] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337–1342, 2003.
- [10] E. Salvador, A. Cavallaro, and T. Ebrahimi, "Cast shadow segmentation using invariant color features," *Computer Vision and Image Understanding*, vol. 95, no. 2, pp. 238–259, 2004.
- [11] D. Grest, J. M. Frahm, and R. Koch, "A color similarity measure for robust shadow removal in real time," in *Proceedings of the Vision, Modeling and Visualization (VMV '03)*, pp. 253–260, November 2003.
- [12] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '99), Frame Rate Workshop*, 1999.
- [13] S. Nadimi and B. Bhanu, "Physical models for moving shadow and object detection in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1079–1087, 2004.
- [14] J. B. Huang and C. S. Chen, "Moving cast shadow detection using Physics-based features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 2310–2317, June 2009.
- [15] N. Martel-Brisson and A. Zaccarin, "Kernel-based learning of cast shadows from a physical model of light sources and surfaces for low-level segmentation," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.
- [16] C. C. Chen and J. K. Aggarwal, "Human shadow removal with unknown light source," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 2407–2410, August 2010.
- [17] J. W. Hsieh, W. F. Hu, C. J. Chang, and Y. S. Chen, "Shadow elimination for effective moving object detection by Gaussian shadow modeling," *Image and Vision Computing*, vol. 21, no. 6, pp. 505–516, 2003.
- [18] H. Nicolas and J. M. Pinel, "Joint moving cast shadows segmentation and light source detection in video sequences," *Signal Processing: Image Communication*, vol. 21, no. 1, pp. 22–43, 2006.
- [19] A. Leone and C. Distanto, "Shadow detection for moving objects based on texture analysis," *Pattern Recognition*, vol. 40, no. 4, pp. 1222–1233, 2007.
- [20] W. Zhang, X. Z. Fang, X. K. Yang, and Q. M. J. Wu, "Moving cast shadows detection using ratio edge," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1202–1213, 2007.
- [21] M. Xiao, C. Z. Han, and L. Zhang, "Moving shadow detection and removal for traffic sequences," *International Journal of Automation and Computing*, vol. 4, no. 1, pp. 38–46, 2007.
- [22] J. Stauder, R. Mech, and J. Ostermann, "Detection of moving cast shadows for object segmentation," *IEEE Transactions on Multimedia*, vol. 1, no. 1, pp. 65–76, 1999.
- [23] M. S. Andersen, T. Jensen, and C. B. Madsen, "Estimation of dynamic light changes in outdoor scenes without the use of calibration objects," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 4, pp. 91–94, August 2006.
- [24] D. S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 827–832, 2005.
- [25] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.
- [26] M. Xu and T. Ellis, "Colour-invariant motion detection under fast illumination changes," in *Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems*, vol. 4, pp. 101–112, September 2001.
- [27] J. Pilet, C. Strecha, and P. Fua, "Making background subtraction robust to sudden illumination changes," in *Proceedings of the 10th European Conference on Computer Vision (ECCV '08)*, pp. 567–580, October 2008.
- [28] J. Choi, H. J. Chang, Y. J. Yoo, and J. Y. Choi, "Robust moving object detection against fast illumination change," *Computer Vision and Image Understanding*, vol. 116, no. 2, pp. 179–193, 2012.

Research Article

Low-Resolution Tactile Image Recognition for Automated Robotic Assembly Using Kernel PCA-Based Feature Fusion and Multiple Kernel Learning-Based Support Vector Machine

Yi-Hung Liu,¹ Yu-Tsung Hsiao,¹ Wei-Teng Cheng,¹ Yan-Chen Liu,² and Jui-Yiao Su²

¹ Department of Mechanical Engineering, Chung Yuan Christian University, Chungli 32023, Taiwan

² Mechanical and Systems Research Laboratories, Industrial Technology Research Institute, Hsinchu 31040, Taiwan

Correspondence should be addressed to Jui-Yiao Su; camussu@itri.org.tw

Received 14 November 2013; Accepted 26 December 2013; Published 24 February 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Yi-Hung Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we propose a robust tactile sensing image recognition scheme for automatic robotic assembly. First, an image reprocessing procedure is designed to enhance the contrast of the tactile image. In the second layer, geometric features and Fourier descriptors are extracted from the image. Then, kernel principal component analysis (kernel PCA) is applied to transform the features into ones with better discriminating ability, which is the kernel PCA-based feature fusion. The transformed features are fed into the third layer for classification. In this paper, we design a classifier by combining the multiple kernel learning (MKL) algorithm and support vector machine (SVM). We also design and implement a tactile sensing array consisting of 10-by-10 sensing elements. Experimental results, carried out on real tactile images acquired by the designed tactile sensing array, show that the kernel PCA-based feature fusion can significantly improve the discriminating performance of the geometric features and Fourier descriptors. Also, the designed MKL-SVM outperforms the regular SVM in terms of recognition accuracy. The proposed recognition scheme is able to achieve a high recognition rate of over 85% for the classification of 12 commonly used metal parts in industrial applications.

1. Introduction

In an automated assembly line, information of object (e.g., shape and orientation) is necessary in the robotic manipulation. Based on the information received, a robot can assemble the products using the objects or parts in an automated manner. Previously, vision-based sensing technique (e.g., CCD camera) was often applied to recognize the shape and orientation information of objects in an automated manufacturing line. Although this approach can provide good temporal and spatial resolutions of objects, its recognition accuracy is easily affected by the environment factors such as lighting conditions. When a robot is operated in a dark environment, the visual sensing quality becomes poor. On the contrary, the visual sensing approach may suffer from the light reflection when the environment becomes brighter, especially when the objects to be assembled are made of metal. Moreover, the objects are sometimes hidden from the visual sensors during the manipulation. In contrast, tactile sensing is less

sensitive to these conditions. Therefore, tactile image-based object recognition has received increasing attention from researchers and engineers over the past decade [1–7].

When the tactile sensing approach is adopted, a two-dimensional tactile sensing array consisting of multiple sensing elements is attached to a robotic hand or finger. When the robotic finger touches an object, each sensing element in the tactile array measures the contact force or pressure applied on a specific and small area of the object. The pressure values of the sensing elements are then transformed into integer ones within the range of [0, 255], thus forming a pseudoimage in which the gray-level values are the transformed pressure values. Based on the pseudoimage (known as the tactile image), a system can recognize the shape, edge direction, and contour of the object. The recognition results are the inputs to the robot which performs the task of automated assembly.

Previous works mainly solve the problem where the size of object is larger/much larger than the tactile sensing array by way of edge tracking/following [1–8]. First, a system/robot

guides the sensing array to move along the contour of an object. After scanning the whole contour of the object, the system estimates the shape of the object by using the information of the collected tactile images during the contour scanning. In addition to this case, there is another: the object is smaller than the tactile sensing array. In this case, the shape of an object can be identified through the use of one single tactile image. However, due to the following factors, the task of object shape recognition becomes difficult and challenging.

- (1) *Low Resolution.* Usually, the size of a tactile sensing array which can be attached to a robotic finger is small, which means that the number of tactile sensing elements in the array is very limited. In other words, the acquired tactile image is of low spatial resolution.
- (2) *Diffusion Effect.* In order to project the tactile sensor array, a thin cover (plastic or silica) is usually placed on the array. Although the cover can prevent the tactile array from being damaged when the robotic finger presses an object, the force that the object applies to one sensing element spreads to its neighboring sensing elements simultaneously. The diffusion effect makes the acquired tactile images ambiguous, especially the edges of the objects.
- (3) *Fence Effect.* Unlike the sensing elements in a CCD camera where the sensing elements are closely adjacent to each other, there exists a large and noticeable gap between sensing elements in a tactile sensing array. These gaps make the object in a sensing tactile image look like fenced.

Due to the factors above, it is difficult to identify the shape of an object through the tactile image. To achieve a high-reliability automated robotic assembly, it is thus necessary to develop a high-accuracy tactile image recognition scheme. To the end, we propose in this paper a scheme, which is composed of three main layers. Initially, an image preprocessing procedure is performed to enhance the contrast of the tactile images. In this layer, geometric features and Fourier descriptors are first extracted from a given image. The extracted geometric features and the Fourier descriptors form a feature vector, which is high-dimensional and does not necessarily achieve satisfactory recognition accuracy. Kernel principal component analysis (kernel PCA) [9] is a powerful kernel method for pattern representation. It computes higher order statistics among random variables while reducing the data dimensionality, thus being able to achieve the goal of both feature extraction and dimensionality reduction. Kernel PCA has shown success in various pattern recognition problems, such as face recognition [10] and defect inspection [11]. Therefore, in this paper, we apply the kernel PCA to reduce the dimensionality and extract more discriminating features from the feature vector extracted from the tactile images. Finally, in the third layer, support vector machine (SVM) [12] is performed to recognize the shape of the object in the input tactile image. In order to improve the generalization performance of SVM, we introduced the multiple kernel learning (MKL) algorithm [13] to the regular SVM. Regular

SVM uses only one single kernel to learn the classifier. Users have to determine the kernel function type and its optimal parameter, which is not only time consuming but also suboptimal because one single kernel may not lead to satisfactory recognition accuracy, especially when the classification problem is complex. Instead of using one single kernel, MKL proposes that an ideal kernel should be a combination of multiple kernels (i.e., base kernels). Based on this idea, MKL trains an SVM with a mixed kernel. In this study, we combine the MKL and the SVM to training a robust classifier for tactile image recognition. Experimental results show that even though the tactile images are of low resolution and suffer from the diffusion and fence effects, the proposed scheme is still able to achieve a high recognition accuracy of over 85% on 12 types of objects.

The rest of this paper is organized as follows. In Section 2, the details of the designed tactile sensing array used in this study as well as the tactile image collection procedure are given. The proposed tactile image recognition scheme is introduced in Section 3. Results and discussion are provided in Section 4. Finally, we conclude this study in Section 5.

2. Tactile Sensor and Image Acquisition

2.1. Material and Manufacturing Process. The piezoresistive layer of sensor is a functional material [14, 15]. It is manufactured by mixing the nanoparticles of carbon and silica into an insulation polymer matrix with high concentration. In this study, we design and fabricate a flexible tactile sensor array by means of screen printing technology. The sensor structure is composed of two 100 μm thick substrates of Polyethylene Terephthalate (PET) films (double-sided electrode structure) and one adhesion layer. Screen printing process is conducted to deposit the silver ink, piezoresistive ink, and adhesion resin on the PET films, respectively, as illustrated in Figure 1. We also design the raise structure to enhance the sensitivity of the tactile sensor. The raised structure is fabricated on the face sheet of the sensor using a UV curable adhesive and has led to high sensitivity due to stress concentration. Similar to the sensing elements on modern touchscreens, these cells can be used for contact localization and pressure measurement. The developed tactile sensor array contains 100 taxels (i.e., 100 tactile sensing elements) in a 10×10 configuration as shown in Figure 2.

2.2. Characteristics of Tactile Sensor Cells. The pressure-piezoresistivity characteristics of the proposed sensor were measured by a customized instrument developed in LabVIEW environment which includes a pressure chamber, multifunction switch/measure unit (Agilent 34980A), and a National Instruments data acquisition (NI DAQ) card. In the calibrating process, the sensor was placed in the chamber, subjected to a static uniform load (to make sure each cell faced the same pressure). The pressures in chamber were controlled by a LabVIEW interface, and the measured data was scanned by Agilent 34980A and recorded via the NI DAQ card. Figure 3 shows the measuring device and a measuring result of one tactile sensor cell. The pressure range

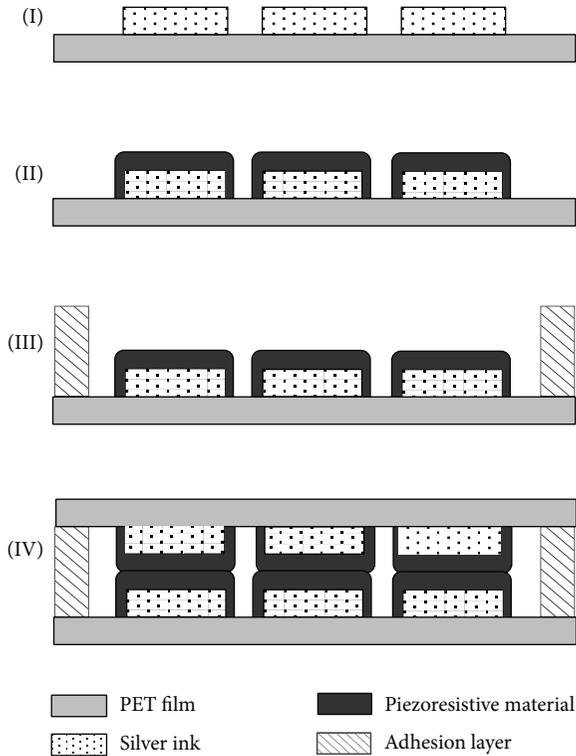


FIGURE 1: Fabrication processes of tactile sensor arrays using screen printing technology. (I) Print the row and column electrodes on the PET films, respectively. (II) Print the piezoresistive material. (III) Bottom PET film with adhesion resin. (IV) The top and bottom PET films are laminated into a large area tactile array sensor.

of the tactile sensor was measured from 10 to 580 psi and the linear relationship between pressure and conductivity in each cell is acceptable.

2.3. Design of Experiment for Image Acquisition. To determine the characteristic of the sensor cell, the experimental setting in Figure 4 was used. It mainly consists of a controlled linear actor using a linear motion stage in conjunction with a high precision servo motor and a load cell. The testing object is placed between the linear actuator's indenter and the tactile sensor array. By positioning the indenter, a defined force can be applied to the testing object, due to the defined flexibility of the load cell. For each sensing element, the sensor material changes its resistance itself as the normal stress changes. The resistivity is measured using a data acquisition (DAQ) module for signal transduction. Each sensing element then provides signals carrying information about the local value of the normal stresses, thus forming a 10×10 tactile image of the contacting object. In this paper we use the tactile image as input to a pattern recognition system, which should classify the contacting object according to its image features.

In addition, to justify the stress applied to collect appropriate tactile images, five different loads (1 kgf, 2 kgf, 3 kgf, 4 kgf, and 5 kgf) were generated by the indenter. Raw images of a bar shape object with fixed cover under the five different loads are shown in Figure 5. The experimental results show

the following: (1) while applying small force (1~3 kgf), the tactile images of the testing object are usually incomplete, due to the uneven press on the contact surface; (2) on the other hand, due to the elastic cover layer, the tactile data present high distortion while applying a too heavy load. According to the observations above, the force of 4 kgf is a suitable one for collecting tactile images.

The contact behavior is mainly determined by the surface flatness and roughness between two objects. That is, local stress will be concentrated on the first contacting area, and this phenomenon will lead to fragment of tactile image. To avoid this phenomenon, we place an elastic cover on a tactile sensor as buffer layer. Several commercially available cover layers with similar hardness were examined under the loading of 4 kgf, as shown in Figure 6. The experimental results show that thicker cover decreases both spatial and force resolutions. However, if the cover is too thin, the tactile image will be more fragile. Accordingly, the 0.9 mm thick silicone sheet was chosen as the cover in this study. Even though the chosen cover is more suitable than others, the diffusion effect mentioned in Section 1 would still happen.

Moreover, the tactile sensor array is fabricated on a flexible film ($20 \text{ mm} \times 20 \text{ mm} \times 250 \mu\text{m}$) and contains 100 sensing elements. Each element has a $1.3 \text{ mm} \times 1.3 \text{ mm}$ sensing area and a 0.7 mm width spacer in each direction. The layout of the designed tactile sensing array is shown in Figure 7. The spacer refers to the gap between elements. As mentioned in Section 1, such a large spacer results in the fence effect.

2.4. Object Types. In this study, 12 mental objects with different shapes and sizes are designed as the testing objects. Samples of the designed objects are shown in Figure 8, and their descriptions are listed in Table 1. These objects are the parts commonly used in the manufacturing and are smaller than the tactile array. When an object is placed on the tactile array, a 4 kgf force is applied on the object. Notice that the cover is placed between the object and the tactile array. When the corresponding tactile image is acquired, the same object is re-put on the tactile array with a slightly different position and direction (orientation) in order to get another image for the same object. By repeating this procedure 20 times, we collect 20 images for each object. Therefore, in this study, the number of classes is 12, and for each class we prepare 20 tactile images. Figure 9 displays examples of the tactile images of the 12 different objects.

As can be observed from these examples, the spatial resolution of the tactile image is extremely low, and it is very difficult to discriminate between objects by observation. For example, the object in the last image (i.e., the sixth image) of the first row and the one in the last image of the second row are originally different: the former is a solid hexagon, while the latter is a solid circle. However, due to the low resolution and the aforementioned diffusion and fence effects, the two different objects in the two images look very similar and are thus difficult to discriminate. Therefore, a robust recognition scheme is required. In the following, we introduce our recognition scheme in detail.

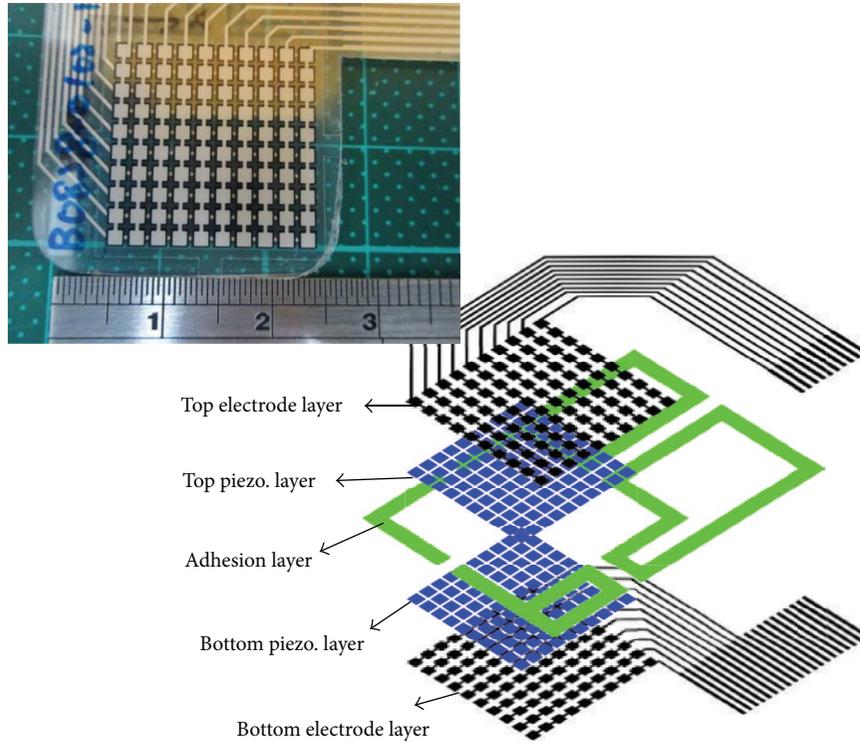


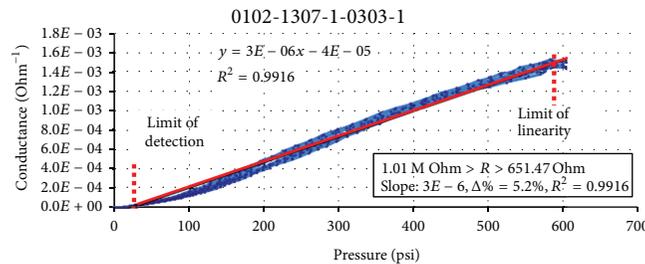
FIGURE 2: Tactile sensor array was fabricated on a flexible film (20 mm × 20 mm × 250 μm) and contains 100 sensor cells (1.3 mm × 1.3 mm).



(a)



(b)



(c)

FIGURE 3: Sensor array calibration device and the multifunction switch/measure unit. (a) The pressure chamber. (b) Agilent 34980A. (c) The pressure-piezoresistivity characteristics of one cell (position (3, 3)).

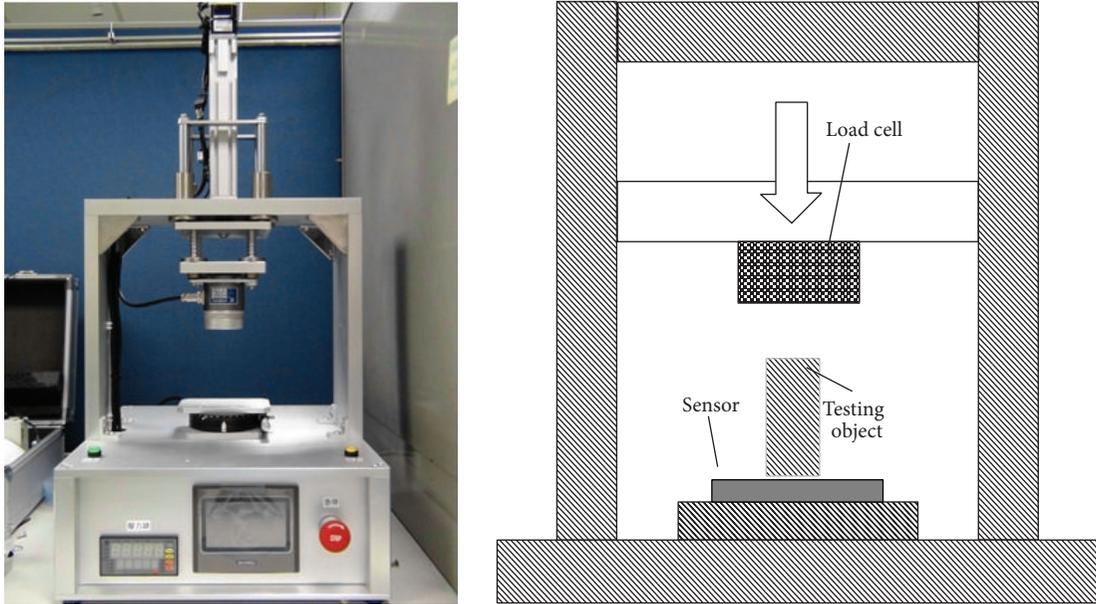


FIGURE 4: Pressure testing machine and experimental setting.

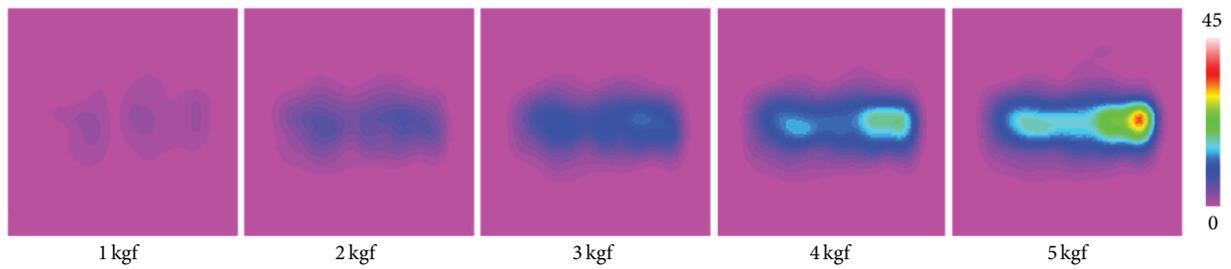
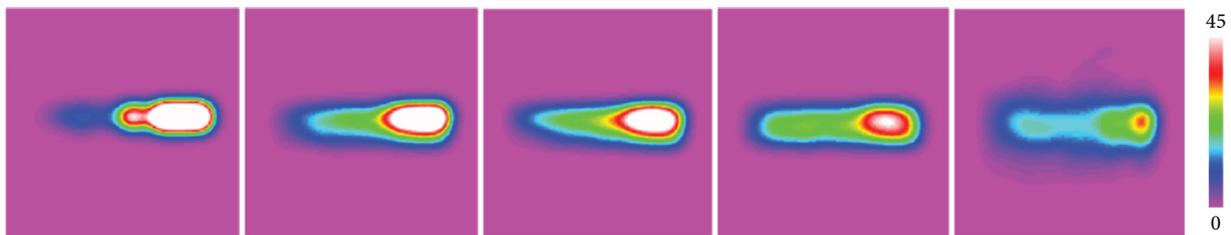


FIGURE 5: Raw images of a bar shape object with fixed cover under various loads 1 kgf~5 kgf.



(a)



(b)

FIGURE 6: Raw images of a bar shape object with various covers under fixed load 4 kgf.

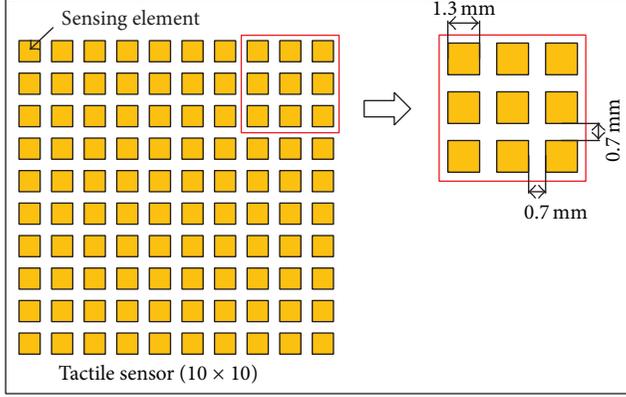


FIGURE 7: Layout of the designed tactile sensor array.

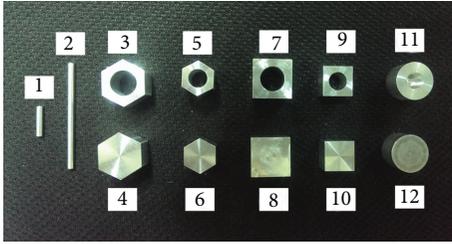


FIGURE 8: Samples of the 12 objects to be recognized in this study.

3. Proposed Tactile Image Recognition Scheme

3.1. Layer 1: Image Preprocessing. Each tactile image is originally a pixel matrix of 10×10 . One example is shown as Figure 10(b). In order to increase the spatial resolution, each image is resized to a 33×33 image by linear interpolation. The resized image of Figure 10(b) is shown as Figure 10(c). As can be observed from Figure 10(c), the resized image has a low contrast. Therefore, Gamma correction [15] is further applied to enhance the contrast of each resized image (see Figure 10(d)). However, the gray levels of pixels near the edge of object and the ones of few isolated pixels inappropriately become higher. To eliminate such noises, a statistical filtering method is performed. Let μ and sd be the mean and the standard deviation of the gray levels of the image. Then, the gray level of one pixel is replaced by zero if the gray level of this pixel is below the threshold T , where $T = \mu - 2 \times sd$. The image after the statistical filtering-based noise removal is shown as Figure 10(e). Finally, the processed gray-level image is transformed to a binarized image through Otsu thresholding [16]. It should be noticed here that when the force is applied to the test object, the force is not necessarily uniformly distributed over the tactile sensing array. Therefore, directly performing the thresholding on the entire image may not lead to an ideal binarized image. If the applied force is highly nonuniformly distributed over the contact surface between the tactile sensing array and the test object, only part of the object will appear after thresholding the entire image according to our preliminary test on the images collected. Thus, our strategy is to partition the image into

TABLE 1: Descriptions of the 12 objects.

Object number	Description
1	Bar shape with 10 mm length
2	Bar shape with 35 mm length
3	Hexagon with flat size of 13 mm and a $\Phi 8$ mm hollow hole
4	Solid hexagon with flat size of 13 mm
5	Hexagon with flat size of 10 mm and a $\Phi 6$ mm hollow hole
6	Solid hexagon with flat size of 10 mm
7	Square with flat size of 13 mm and a $\Phi 8$ mm hollow hole
8	Solid square with flat size of 13 mm
9	Square with flat size of 10 mm and a $\Phi 6$ mm hollow hole
10	Solid square with flat size of 10 mm
11	$\Phi 13$ mm circle with a $\Phi 8$ mm hollow hole
12	Solid $\Phi 13$ mm circle

$n \times n$ subimages with equal size and then perform the Otsu thresholding on each subimage independently. We found that when n is set as 2, the highest recognition accuracy can be obtained. The binarized results of 1×1 , 2×2 , and 3×3 are shown in Figures 10(f), 10(g), and 10(h), respectively.

3.2. Layer 2: Kernel PCA-Based Feature Fusion

3.2.1. Geometric Features. Two kinds of geometric features are extracted from each binarized image: area and edge-to-mean variance (called variance hereafter). Area denotes the number of pixels labeled as 1 in the binarized image. To compute the variance, we first detect the edge points and the centroid of the object within one binarized image and then compute the distance between each edge point and the centroid. Finally, the variance of the computed distances is calculated.

3.2.2. Fourier Descriptors. To compute the Fourier descriptors, the boundary extraction algorithm [17] is performed to find the Cartesian coordinates of the sequential boundary pixels of the object in an image. Examples of the images after the boundary extraction are shown in Figure 11.

Suppose that the Cartesian coordinates of the boundary pixels of an object are $(x[m], y[m])$, $m = 1, 2, \dots, L$, where L denotes the number of boundary pixels. The Fourier series expansion of the boundary pixels is as follows:

$$x[m] = \sum_{n=-\infty}^{\infty} a[n] e^{jn\omega_0 m}, \quad (1a)$$

$$y[m] = \sum_{n=-\infty}^{\infty} b[n] e^{jn\omega_0 m}, \quad (1b)$$

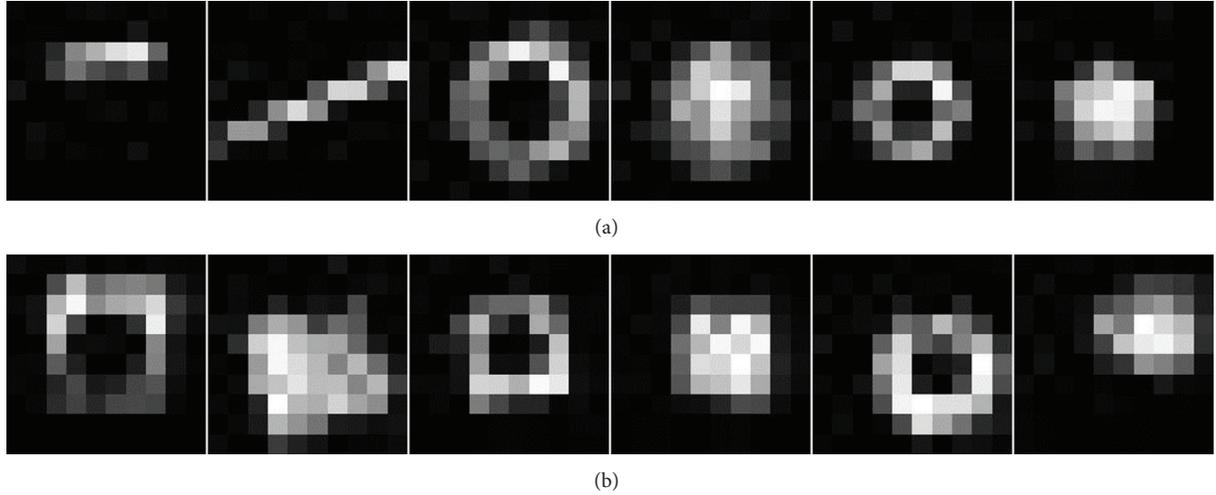


FIGURE 9: Examples of the tactile images. The images in the first row of this figure are the tactile images of the first six objects (class 1–class 6), respectively. The second row displays the examples of the tactile images of class 7–class 12, respectively. Each image is a 10-by-10 gray-level matrix.

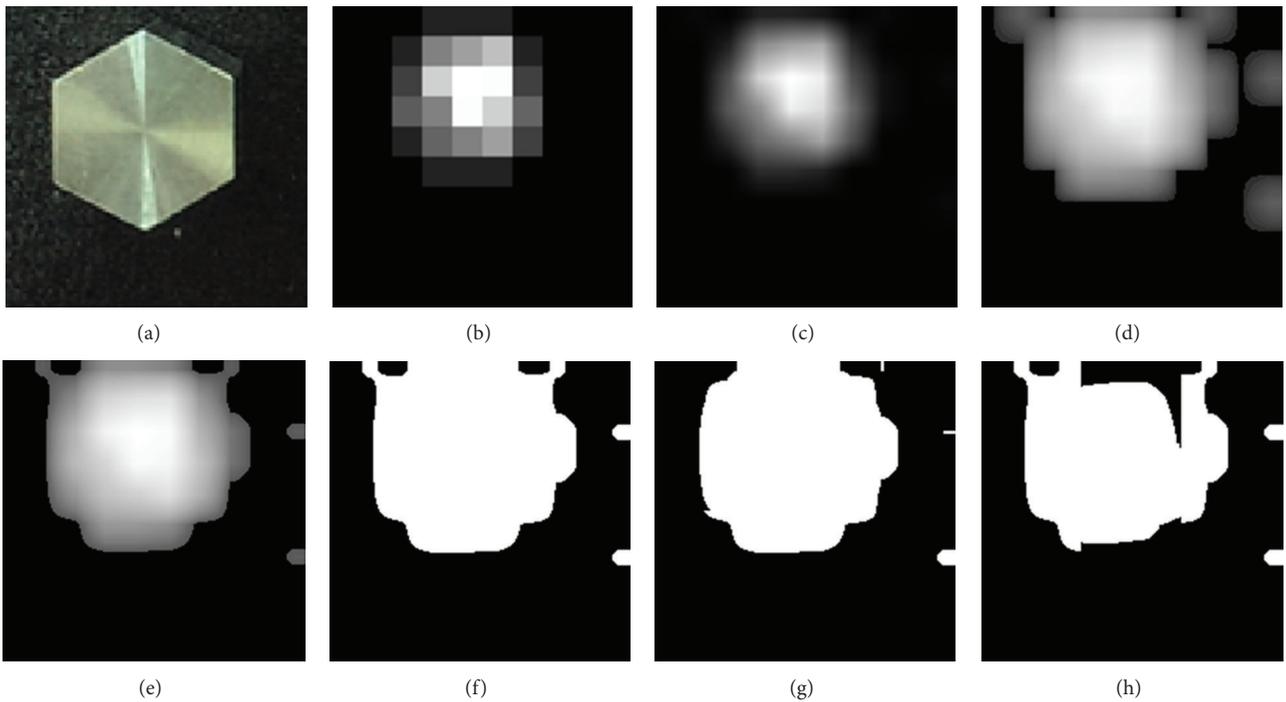


FIGURE 10: An illustrative example for the image preprocessing stage, where (a) is the testing object, (b) is the corresponding 10-by-10 tactile image, (c) is the resized 33-by-33 image, (d) is the image after Gamma correction-based contrast enhancement, (e) is the image after noise reduction, and (f)–(h) are the binarized images with 1×1 , 2×2 , and 3×3 image partitioning-based thresholding.

where $\omega_0 = 2\pi/L$ and $a[n]$ and $b[n]$ are Fourier coefficients:

$$a[n] = \frac{1}{L} \sum_{m=1}^L x[m] e^{-jn\omega_0 m}, \quad n = 1, \dots, L, \quad (2a)$$

$$b[n] = \frac{1}{L} \sum_{m=1}^L y[m] e^{-jn\omega_0 m}, \quad n = 1, \dots, L. \quad (2b)$$

The Fourier descriptors $s[n]$ are given by

$$s[n] = \frac{r[n]}{r[1]}, \quad n = 1, \dots, L, \quad (3)$$

where $r[n] = \sqrt{|a[n]|^2 + |b[n]|^2}$. The Fourier descriptors are translation-, rotation-, and scaling-invariant. Actually, not all of the L Fourier descriptors are required: only the first n descriptors are necessary and perform the best. After

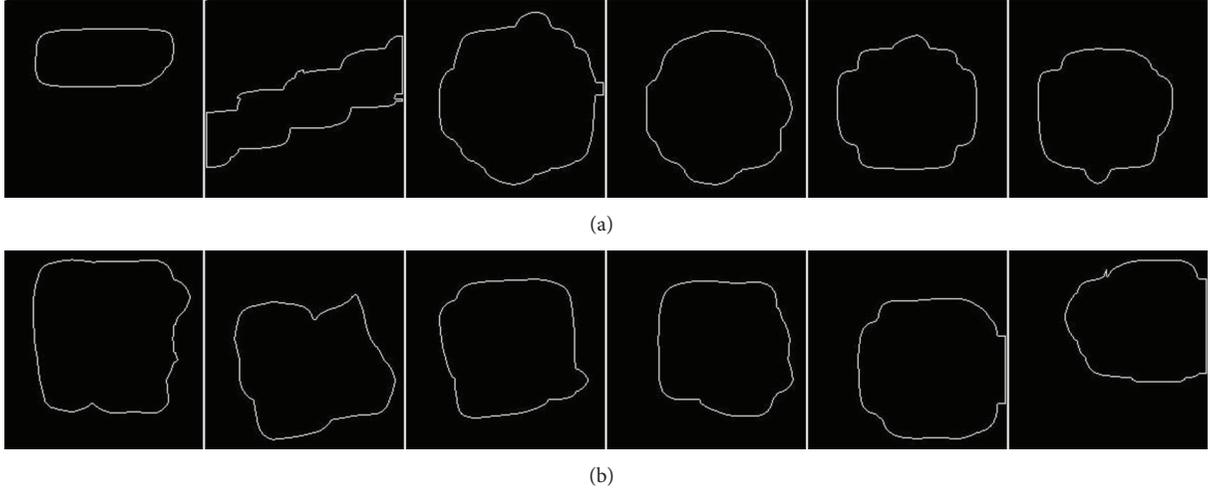


FIGURE 11: Examples of boundary extraction result. The corresponding gray-level tactile images are displayed in Figure 9.

the feature extraction, each tactile image is represented by a vector of $n + 2$ dimension, in which two are geometric features and the rest are the extracted Fourier descriptor. To facilitate the following illustration, the feature vectors are simply called data hereafter.

3.2.3. Kernel PCA-Based Feature Transformation. The kernel PCA feature fusion consists of a training phase and a testing phase. Suppose that there is a set of training data $\mathbf{x}_i \in R^m$, $i = 1, \dots, M$. Kernel PCA maps the training data into a higher-dimensional feature space F using a nonlinear mapping $\phi : R^m \rightarrow F$, where $m = n + 2$, and then centers these mapped data such that they have a zero mean: $\sum_{i=1}^M \phi(\mathbf{x}_i) = \mathbf{0}$ (see [9] for a detailed derivation of the data centering method in the feature space). In the feature space, kernel PCA solves the following eigenvalue problem:

$$\lambda \mathbf{v} = \Gamma \mathbf{v}, \quad (4)$$

where $\mathbf{v} \in F$ are eigenvectors associated with nonzero eigenvalues λ and $\Gamma = 1/M(\sum_{i=1}^M \phi(\mathbf{x}_i)\phi^T(\mathbf{x}_i))$ is the mapped-data covariance matrix. By introducing the kernel function: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$, the dual problem of (1a) and (1b) is as follows:

$$M\lambda \mathbf{a} = \mathbf{K} \mathbf{a}, \quad (5)$$

where $\mathbf{K} : K_{ij} \equiv K(\mathbf{x}_i, \mathbf{x}_j)$ is a $M \times M$ kernel matrix and $\mathbf{a} = (a_1, \dots, a_M)^T$ is the eigenvector associated with $\lambda_i \neq 0$ and is subject to the normalization condition $\|\mathbf{a}\|^2 = 1/\lambda$. Solving the eigenvalue problem expressed as (5) yields M eigenvectors \mathbf{a}^k , $k = 1, \dots, M$. However, we select only the first d leading eigenvectors as the basis for transformation. The number of chosen eigenvector should be smaller than the number of total features and the number of training data, that is, $d < m$ and $d < M$, and the optimal number of the eigenvectors should be experimentally determined. By doing so, the goal of dimensionality reduction can be achieved.

After the eigenvector selection, the training phase of kernel PCA is completed.

In test phase, the projection of testing data $\mathbf{x} \in R^m$ onto the k th eigenvector \mathbf{v}^k is computed by

$$z^k = \mathbf{v}^k \cdot \phi(\mathbf{x}) = \sum_{i=1}^M a_i^k K(\mathbf{x}_i, \mathbf{x}), \quad k = 1, \dots, d, \quad (6)$$

where a_i^k is the i th component of the k th eigenvector \mathbf{a}^k , and z^k is the nonlinear principal component of \mathbf{x} corresponding to the nonlinear mapping ϕ . The d nonlinear principal components constitute a vector $\mathbf{z} = (z^1, \dots, z^d)^T$, which is the nonlinear fusion of the geometric features and the Fourier descriptors. In this study, the Gaussian function $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/2\sigma^2)$ is chosen as the kernel, where σ is a user-specified kernel parameter and can be optimized by using a cross validation procedure.

3.3. Layer 3: Multiple Kernel-Based SVM Classification

3.3.1. SVM. Given a training set $\{\mathbf{z}_i, y_i\}$, $i = 1, \dots, M$, where $\mathbf{z}_i \in R^d$ are training data and $y_i \in \{-1, +1\}$ are class labels, SVM maps the data into a higher-dimensional feature space and then finds an optimal separating hyperplane (OSH) which maximizes the margin of separation and minimizes the training errors simultaneously, which can be formulated as the constrained optimization problem as

$$\begin{aligned} & \text{Minimize} && \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^M \xi_i \\ & \text{subject to} && y_i (\mathbf{w}^T \phi(\mathbf{z}_i) + b) - 1 + \xi_i \geq 0, \quad \forall i \\ & && \xi_i \geq 0, \quad \forall i, \end{aligned} \quad (7)$$

where \mathbf{w} denotes the weight vector of the hyperplane, b is the bias of the hyperplane, ξ_i are slack variables representing training errors, and C is a penalty weight. The value of C needs

to be specified in prior. Introducing the Lagrangian to (7) yields the dual problem:

$$\begin{aligned} & \text{Maximize} \quad \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^M \alpha_i \alpha_j y_i y_j K(\mathbf{z}_i, \mathbf{z}_j) \\ & \text{subject to} \quad 0 \leq \alpha_i \leq C, \quad \forall i \\ & \quad \quad \quad \sum_{i=1}^M \alpha_i y_i = 0, \end{aligned} \quad (8)$$

where α_i are Lagrange multipliers. The training data for which $0 < \alpha_i \leq C$ are called support vectors (SVs). The class label for a test data \mathbf{z} is computed by the decision function:

$$D(\mathbf{z}) = \text{Sign} \left(\sum_{\mathbf{z}_i \in \text{SV}} \alpha_i y_i K(\mathbf{z}_i, \mathbf{z}) + b_o \right), \quad (9)$$

where b_o is the optimal bias of the OSH, which can be calculated by taking any support vectors whose corresponding Lagrange multipliers satisfy $0 < \alpha_i < C$ into the Kuhn-Tucker conditions. If $D(\mathbf{z}) > 0$, \mathbf{z} is classified as a positive data or negative data otherwise.

3.3.2. Multiple Kernel SVM. MKL is a data-driven learning algorithm which learns kernel from the given training data [13]. It assumes that an ideal kernel is a linear combination of predefined base kernels:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^N d_k K_k(\mathbf{x}_i, \mathbf{x}_j), \quad (10)$$

where K_k are base kernels, d_k are kernel combination weights, and N is the number of chosen base kernels. MKL-SVM solves the following optimization problem:

$$\begin{aligned} & \text{Minimize} \quad \sum_{k=1}^N \frac{1}{2} \frac{\|\mathbf{w}_k\|^2}{d_k} + C \sum_{i=1}^M \xi_i \\ & \text{subject to} \quad y_i \left(\sum_{k=1}^N \mathbf{w}_k^T \phi_k(\mathbf{z}_i) + b \right) - 1 + \xi_i \geq 0, \quad \forall i, \\ & \quad \quad \quad \xi_i \geq 0, \quad \forall i, \\ & \quad \quad \quad \sum_{k=1}^N d_k = 1, \quad d_k \geq 0, \quad \forall k. \end{aligned} \quad (11)$$

The dual problem of (11) is written as

$$\begin{aligned} & \text{Maximize}_{\alpha, \gamma} \quad \sum_{i=1}^M \alpha_i - \gamma \\ & \text{subject to} \quad 0 \leq \alpha_i \leq C, \quad \forall i, \\ & \quad \quad \quad \sum_{i=1}^M \alpha_i y_i = 0, \\ & \quad \quad \quad \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^M \alpha_i \alpha_j y_i y_j K_k(\mathbf{z}_i, \mathbf{z}_j) \leq \gamma, \quad \forall k. \end{aligned} \quad (12)$$

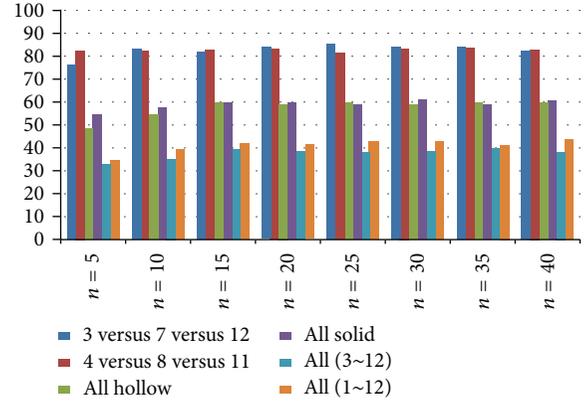


FIGURE 12: k -NN recognition accuracies of Fourier descriptors among different class combinations.

Finally, for a test data point \mathbf{z} its class label is determined by the MKL-SVM decision function:

$$f(\mathbf{z}) = \text{Sign} \left(\sum_{i=1}^M \sum_{k=1}^N \alpha_i y_i d_k K_k(\mathbf{z}_i, \mathbf{z}) + b \right). \quad (13)$$

In this paper, we solve the optimal values of α_i , d_k , and b by using the Simple MKL by Rakotomamonjy et al. [18] for it adopts the reduced gradient method to solve the MKL optimization problem, which is computationally cheaper than other MKL solvers.

4. Results and Discussion

In this section, we first test the recognition accuracies of the geometric features or properties (GP) and Fourier descriptors (FD) using a simple classifier, that is, the k -nearest neighbor (k -NN) classifier, to find the optimal number of FDs, where k is set as 3. A ten-run twofold cross validation procedure is performed to test the recognition accuracy on the data set. The data set contains 240 data belonging to 12 classes, and each class has 20 data. The results are shown in Figures 12–14. Notice that (3 versus 7 versus 12) means that the number of objects to be classified is 3, including class 3, class 7, and class 12. Also, “all hollow” means that the objects to be classified are the ones with hollow holes as shown in Figure 8, including class 3, class 5, class 7, class 9, and class 11, totally 5 classes. On the other hand, “all solid” contains 5 different classes, including class 4, class 6, class 8, class 10, and class 12. “All (1–12)” means that the number of classes is 12 (from class 1 to class 12).

It can be seen from Figure 12 that different class combination results in different accuracy. For example, in the case of “All solid,” FD with $n = 40$ gives a recognition accuracy of 60.8%. For the same case, GP results in recognition accuracy of 74.45% (see Figure 13). When the two kinds of features are combined, a higher recognition accuracy of 84.3% occurs, as shown in Figure 14. Moreover, if the objective is to classify all types of objects, that is, the case of All (1–12), the recognition accuracy of FD with $n = 40$ is only 43% (see Figure 12), and the GE gives only the recognition accuracy

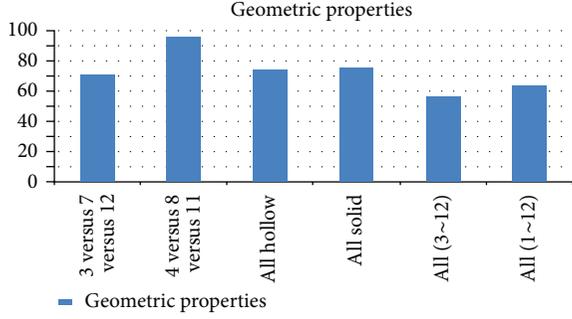


FIGURE 13: k -NN recognition accuracies of geometric features among different class combinations.

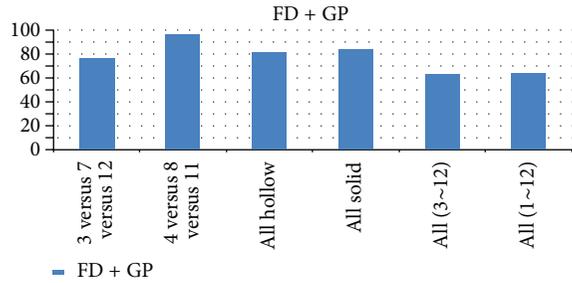


FIGURE 14: k -NN recognition accuracies of geometric features and Fourier descriptors among different class combinations.

of 63.3 (see Figure 13). However, when the two kinds of features are combined, the accuracy can be enhanced to 68.69%, as indicated in Figure 14. These comparisons show that the combination of GE and FD is able to achieve higher recognition accuracy than any of them. In addition, we can also observe from Figure 12 that the recognition accuracy saturates as n reaches 20. Moreover, when $n = 40$, FD gives the best result. Therefore, we set $n = 40$ in the following experiments.

Next, we test the proposed recognition scheme (combination of the kernel PCA-based feature fusion and MKL-SVM) and compare the proposed scheme with other combinations. Similarly, the 10-run twofold cross validation is performed to optimize the parameters of the methods. For kernel PCA, the parameters to be optimized include the kernel parameter and the number of eigenvectors. The parameters of SVM are the penalty weight C and the kernel parameter σ . For MKL-SVM, not only the penalty weight C needs to be adjusted, but also the based functions need to be determined in advance. Choosing a set of good kernels as the base kernels is crucial to the MKL-SVM. Accordingly, two kinds of frequently used kernel functions are adopted as the base kernels in this study: Gaussian function and the polynomial function:

$$\begin{aligned} K_G(\mathbf{x}, \mathbf{y}) &= \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / 2\sigma^2), \\ K_P(\mathbf{x}, \mathbf{y}) &= (1 + \mathbf{x}^T \mathbf{y})^p, \end{aligned} \quad (14)$$

where p is the power of the polynomial kernel. In order to get a good combination of kernels, the range of σ is

TABLE 2: Comparison of recognition accuracies among different methods.

Feature	Classifier	Recognition accuracy (in %)
FD + GE	k -NN	68.69
FD + GE	SVM	76.17
Kernel PCA-based feature fusion	SVM	82.13
Kernel PCA-based feature fusion	MKL-SVM	85.54

set as wide as possible: $\sigma = \{2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}\}$. Also, the values of p are in the set of $\{1, 2, 3, 4, 5, 6\}$. There are totally 16 Gaussian kernels and 6 polynomial kernels used in the MKL-SVM. Therefore, the number N of base kernels is 22 in this experiment. In addition, since the objective of this study is to classify 12 different objects, we only consider this case in this experiment. Since there are 12 classes to be classified, the binary classifier SVM and MKL-SVM needs to be extended to multiclass classifiers. In this study, the one-against-one method combined with the voting strategy [19] is adopted for this purpose. Finally, the cross validation results are listed in Table 2.

As can be seen from Table 2, when feature FD + GE is used, SVM largely improves the recognition accuracy. However, the accuracy of 76.17% is still unacceptable when the method is applied to automated robotic assembly. When the kernel PCA is further applied to FD + GE, namely, the kernel PCA-based feature fusion, the recognition accuracy is significantly improved from 76.17% to 82.13%, which demonstrates the validity of the proposed feature fusion scheme in improving the tactile image recognition. Further, when the classifier is replaced by MKL-SVM, an accuracy improvement of 3.41% (85.54% - 82.13%) is observed. Although this difference appears to be small, the error reduction ratio is large: $3.41 / (100 - 82.13) = 19.08\%$. Therefore, we can conclude that MKL-SVM is more suitable than the widely used SVM for tactile image classification.

5. Conclusion

In this paper, we have presented a recognition scheme for solving the difficult tactile image recognition problem, which plays a critical role in automated robotic assembly. The proposed kernel PCA-based feature fusion technique largely improved the recognition accuracy of the frequently used geometric featured and Fourier descriptors, and the multiple kernel learning (MKL)-based SVM can perform much better than the regular SVM in terms of object recognition through the use of tactile image. Experimental results have indicated the effectiveness of the proposed recognition scheme in tactile image recognition. Nevertheless, there remain several worth-studying issues that may further improve the current results. For example, other types of kernels can be included in the MKL-SVM to gain better kernel combination, which will be our future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] T. C. Phung, Y. S. Ihn, J. C. Koo, and H. R. Choi, "Edge identification of a small object through a low-resolution tactile sensor array," *International Journal of Precision Engineering and Manufacturing*, vol. 11, no. 2, pp. 247–254, 2010.
- [2] N. Chen, R. Rink, and H. Zhang, "Efficient edge detection from tactile data," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 386–391, August 1995.
- [3] G. M. Krishna and K. Rajanna, "Tactile sensor based on piezoelectric resonance," *IEEE Sensors Journal*, vol. 4, no. 5, pp. 691–697, 2004.
- [4] K. Suwanratchatamane, M. Matsumoto, and S. Hashimoto, "Robotic tactile sensor system and applications," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 3, pp. 1074–1087, 2010.
- [5] Z. Pezzementi, C. Reyda, and G. D. Hager, "Object mapping, recognition, and localization from tactile geometry," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 5942–5948, Shanghai, China, May 2011.
- [6] Z. Pezzementi, E. Plaku, C. Reyda, and G. D. Hager, "Tactile-object recognition from appearance information," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 473–487, 2011.
- [7] U. M. Hernandez, T. J. Dodd, L. Natale, G. Metta, T. J. Prescott, and N. F. Lepora, "Active contour following to explore object shape with robot touch" in *Proceedings of the IEEE World Haptics Conference (WHC '13)*, pp. 341–346, Daejeon, Republic of Korea, April 2013.
- [8] Z. Pezzementi, E. Jantho, L. Estrade, and G. D. Hager, "Characterization and simulation of tactile sensors," in *Proceedings of the IEEE Haptics Symposium*, pp. 199–205, Waltham, Mass, USA, March 2010.
- [9] B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a Kernel Eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [10] K. I. Kim, K. Jung, and H. J. Kim, "Face recognition using kernel principal component analysis," *IEEE Signal Processing Letters*, vol. 9, no. 2, pp. 40–42, 2002.
- [11] Y. H. Liu, C. K. Wang, Y. Ting et al., "In-TFT-array-process micro defect inspection using nonlinear principal component analysis," *International Journal of Molecular Sciences*, vol. 10, no. 10, pp. 4498–4514, 2009.
- [12] V. N. Vapnik, *Statistical Learning Theory*, Adaptive and Learning Systems for Signal Processing, Communications, and Control, Springer, New York, NY, USA, 1998.
- [13] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.
- [14] K. Weiß and H. Worn, "The working principle of resistive tactile sensor cells," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '05)*, pp. 471–476, Niagara Falls, Canada, August 2005.
- [15] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, New York, NY, USA, 2nd edition, 2003.
- [16] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [17] Y. H. Liu, *Feature analysis and classifier design and their applications in computer vision and data mining [Ph.D. thesis]*, National Taiwan University, Taipei, Taiwan, 2003.
- [18] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.
- [19] Y. H. Liu and Y. T. Chen, "Face recognition using total margin-based adaptive fuzzy support vector machines," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 178–192, 2007.

Research Article

An Improved Adaptive Deconvolution Algorithm for Single Image Deblurring

Hsin-Che Tsai and Jiunn-Lin Wu

Department of Computer Science and Engineering, National Chung Hsing University, Taichung 40227, Taiwan

Correspondence should be addressed to Jiunn-Lin Wu; jlwu@cs.nchu.edu.tw

Received 3 October 2013; Accepted 3 January 2014; Published 19 February 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 H.-C. Tsai and J.-L. Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the most common defects in digital photography is motion blur caused by camera shake. Shift-invariant motion blur can be modeled as a convolution of the true latent image and a point spread function (PSF) with additive noise. The goal of image deconvolution is to reconstruct a latent image from a degraded image. However, ringing is inevitable artifacts arising in the deconvolution stage. To suppress undesirable artifacts, regularization based methods have been proposed using natural image priors to overcome the ill-posedness of deconvolution problem. When the estimated PSF is erroneous to some extent or the PSF size is large, conventional regularization to reduce ringing would lead to loss of image details. This paper focuses on the nonblind deconvolution by adaptive regularization which preserves image details, while suppressing ringing artifacts. The way is to control the regularization weight adaptively according to the image local characteristics. We adopt elaborated reference maps that indicate the edge strength so that textured and smooth regions can be distinguished. Then we impose an appropriate constraint on the optimization process. The experiments' results on both synthesized and real images show that our method can restore latent image with much fewer ringing and favors the sharp edges.

1. Introduction

Image blurring is one of the prime causes of poor image quality in digital photography. The one main cause of blurry images is motion blur caused by camera shake. If a motion blur is linear shift invariant, the blurring process can be generally modeled as a convolution of the true latent image and a point spread function (PSF) with additive noise:

$$B = K \otimes I + N, \quad (1)$$

where B is the degraded image, I is the true latent image, K is the PSF or a motion blur kernel which describes the trace of a sensor, N is the additive noise introduced during image acquisition, and \otimes denotes the convolution operator. The goal of image deblurring is to reconstruct a latent image I from degraded image B .

To remove motion blur, we need to estimate the PSF and restore a latent image through deconvolution. Existing single image deblurring methods can be further categorized into two classes. If both the PSF and the latent image are

unknown, the challenging problem is called blind deconvolution. Although great progresses have been achieved in the recent years [1–4], blind case is severely ill-posed problem because the number of unknowns exceeds the number of observed data. In contrast to the former, if the PSF is assumed to be known or computed in other ways, the problem is reduced to estimating the latent image alone. This is called nonblind deconvolution. However, the nonblind case is still an ill-conditioned problem that has to do with the presence of image noise. Slight mismatches between the PSF used by the method and the true blurring PSF also lead to poor deblurring results.

Unfortunately, the deconvolved result usually contains unpleasant artifacts even if the PSF is exactly known or well estimated. The main visually disturbing artifact is ringing that appears around strong edges. Because the PSF is often band-limited with a sharp frequency cutoff, that will be zero or near-zero values in its frequency response. Thus the direct inverse of the PSF causes large amplification of signal and noise.

Since the estimated PSF is usually inaccurate and the real blurred image is also noisy, many underconstrained factors will affect the amplification of ringing. To reduce undesirable artifacts, various regularization techniques have been proposed using image priors to improve nonblind deconvolution methods [5–10]. The most commonly used priors are those which encourage the image gradients to a set of derivative filters to follow heavy-tailed distribution. However, strong regularization to reduce severe artifacts destroys the image details in the deconvolved result. Weak regularization preserves image details well, but it does not remove artifacts tellingly. The challenge of this work is how to balance the details recovery and ringing suppression.

In this paper, we focus on non-blind deconvolution with adaptive regularization that controls the regularization strength according to the image local characteristics. This strategy reduces ringing artifacts in a smooth region effectively and preserves image details in a textured region simultaneously. First, we estimate the reference maps in the scale space. At the coarsest scale, we are able to extract reasonably main strong edges. Comparing the texture information of multiscale results, we can make the elaborated reference map that differentiates the edge property of textured region and smooth region. Second, regularization strength is controlled adaptively referring to these maps. Then, we apply appropriate regularization with hyper-Laplacian prior to image deconvolution so that the sharp edges can be eventually recovered. To solve the optimization process, we adopt Krishnan's [7] fast algorithm. It is performed fast in the frequency domain using fast Fourier transforms (FFTs). The experimental results show that our nonblind deconvolution can produce latent image with much fewer ringing and preserve the sharp edges.

2. Regularization Formulation

Assuming that the PSF is known or computed in other ways, our method focuses on recovering the sharp image from the blurred image. To reduce undesirable artifacts, most advanced regularization techniques are proposed using the prior of nature image in the gradient domain. We now introduce the regularization formulation. From a probabilistic perspective, we seek maximum a posteriori (MAP) estimate of latent image I in Bayesian framework:

$$p(I | B) \propto p(B | I) p(I), \quad (2)$$

where $p(B | I)$ represents the likelihood and $p(I)$ denotes the priors on the latent image. Maximizing $p(I | B)$ is equivalent to minimizing the cost $-\log p(I | B)$:

$$-\log p(I | B) = -\log p(B | I) - \log p(I). \quad (3)$$

The MAP solution of I can be obtained by minimizing the cost function above. We now define these two terms. The likelihood of a degraded image given the latent image is based on the common blur model $N = B - K \otimes I$. Assuming that the noise is modeled as a set of independent and identically distributed (i.i.d.) noise random variables for all pixels, each

of which follows a Gaussian distribution, we can express the likelihood with Gaussian variance σ^2 as

$$p(B | I) \propto e^{-(1/2\sigma^2)\|B-K\otimes I\|^2}. \quad (4)$$

Furthermore, recent research in natural image statistics shows that image gradients follow a heavy-tailed distribution. These distributions are well modeled by a hyper-Laplacian prior and have proven effective priors for deblurring problem. In this paper, we utilize the hyper-Laplacian prior to regularize the solution and it can be modeled as

$$p(I) \propto e^{-\alpha \sum_{k=1}^2 |f_k \otimes I|^q} \quad \text{with } 0.5 \leq q \leq 0.8, \quad (5)$$

where q is a positive exponent value set in the range of $0.5 \leq q \leq 0.8$ as suggested by Krishnan and Fergus [7]. In this work, we unify the use of $2/3$ for q value. The f_k denotes the simple horizontal and vertical first-order derivative filters. It can also be useful to include second-order derivative filters or the more sophisticated filters. With the Gaussian noise likelihood and the hyper-Laplacian image prior, (3) can be represented by the following minimization problem:

$$\arg \min_I \sum_{i=1}^n \left((B - K \otimes I)_i^2 + \eta \sum_{k=1}^2 |(f_k \otimes I)_i|^q \right), \quad (6)$$

where i is an index running through all pixels, n is the whole pixels in the image, and weighting coefficient $\eta = 2\alpha\sigma^2$ controls the strength of the regularization term. For simplicity, $f_1 = [1 \ -1]$ and $f_2 = [1 \ -1]^T$ are two first-order derivative filters. We search for the I which minimizes the reconstruction error $\|B - K \otimes I\|^2$, with the image prior preferring I to favor the correct sharp explanation.

However, to analyze the formulation of Equation (6), the above conventional regularization's weighting coefficient η is applied to all pixels with the same strength. When the estimated PSF is inaccurate or the PSF size is large, it usually adds strong regularization weight for suppressing ringing around the edges. But it also destroys the image details in the deconvolved result, and this is inevitable problem since perfect PSF estimation is impossible. In addition, the weak regularization weight preserves major details well, but it does not remove artifacts effectively.

3. The Proposed Method

To balance the details recovery and ringing suppression, the main ideal of our approach is to control the regularization weight adaptively according to the image local characteristics. This means that we need strong regularization for the smooth regions and weak regularization for the sharp edges. Our estimated maps further consider the edge property of image textured regions. In this chapter, we will explain each step of our algorithm in detail. That includes how to estimate reference map and perform the adaptive regularization in the deconvolution stage. Finally, we supply some added improvement to obtain better results. Figure 1 shows the overall process of our nonblind deconvolution method.

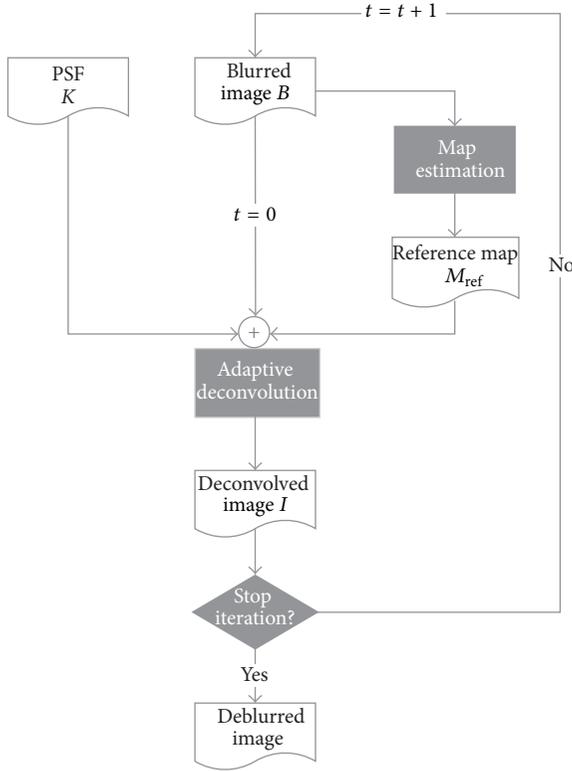


FIGURE 1: Overview of our adaptive deconvolution framework.

3.1. Reference Map Estimation. Inspired by the local constraint idea [8, 9, 11], the reference map is designed for classifying the smooth regions and different edge regions correctly. Reference map is used for deconvolution with adaptive regularization and can apply the right image characteristics information to control the local weight of deblurring. In fact, the classified texture regions also contain the main strong edges and some smaller or fainter edges. Compared to the smaller or fainter edges, the extracted main edges are the important visible regions, and those need more weak regularization to preserve the sharp features. In addition, we should add strong regularization for the classified smooth regions to reduce the most noticeable artifacts since the ringing artifacts are usually propagated in the smooth areas.

Based on these observations, we estimate the reference maps in the multiscale space. We first build a pyramid $\{B^l\}_{l=0}^L$ of the full resolution input blurred image B using bilinear downsampling. Our goal is to combine different edge information from coarse to fine layers. So we perform the map estimation approach in each scale and distinguish its different regions. At the coarsest scale, we are able to extract reasonably main strong edges of the image. In the next finer scale, we gradually extract more and more small edge details. Finally, all estimated maps from different scale results combine to form one elaborated reference map. The estimation is guided to a good map by concentrating on the main edges of the input image and progressively dealing with smaller and smoother details. But it is difficult to obtain

correct edge information from the blurred image directly, so the map is renewed from every deconvolved result during each iteration time.

We now describe how to produce and combine the estimated reference maps. The initial reference map is estimated from the input blurred image. Since the locally smooth region which has no edges information is still smooth after blurring, the edge areas of blurred image will be affected. Inspired by this idea, we compute the edge strength and use predefined threshold to classify different regions. The edge strength at located pixel i is defined as follows:

$$\text{Eg}(i) = \frac{\left(\sum_{h \in W_x} h + \sum_{v \in W_y} v\right)}{N_{\text{total}}}, \quad (7)$$

where $\text{Eg}(i)$ denotes the edge strength response at pixel i on the observed image. W is the 3×3 window whose center is located on i , $W_x = W \otimes [1 \ -1]$, $W_y = W \otimes [1 \ -1]^T$, and N_{total} is the total number of pixels in default window. If the computed edge strength value is smaller than a predefined threshold T , which is set to 3×10^{-2} in our experiment, we will regard the center pixel i as in smooth region Ω , that is, $i \in \Omega$:

$$M_i^l = \begin{cases} 1 & \text{if } \text{Eg}(i) < T \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Since the texture regions which is classified by (7) also contain the main strong edges and some smaller edges, we aim to further improve the estimated results. The improved extracting method includes the following steps. Firstly, we build a pyramid $\{B^l\}_{l=0}^L$ of the blurred image. At each scale l , the edge strength $\text{Eg}(i)$ is computed in every level and then is upsampled. It means that this step will produce l number binary maps. Finally, we sum all the edge intensities at the same pixel location and run through all pixels. Now we get a new reference map after normalization process, which is shown in Figure 2(b), and the smooth region Ω is shown as the set of all white pixels. The strongest edges are indicated by the most dark pixels and the other gray level pixels mean the smaller or fainter edges. Hence, we define this extracting step using the multiscale approach of the observed image as:

$$M_{\text{ref}} = \frac{\sum_{l=0}^L M^l}{L}, \quad (9)$$

where M^l is the binary map at the l th level of the images pyramid by (8), and L denotes the total scale layers, for example, L is set to 3 levers in our implementation. In (9), the final map M_{ref} is extracted by summing up all the intensity values of those upsampling maps and normalizing the results.

We gradually recover more and more image details by an iterative deconvolution algorithm because it can refine the result until convergence. Obviously, the deconvolved image with initial adaptive regularization shows much better edge information than does the input blurred image. Thus, at the following reference map estimation, we use the deconvolved

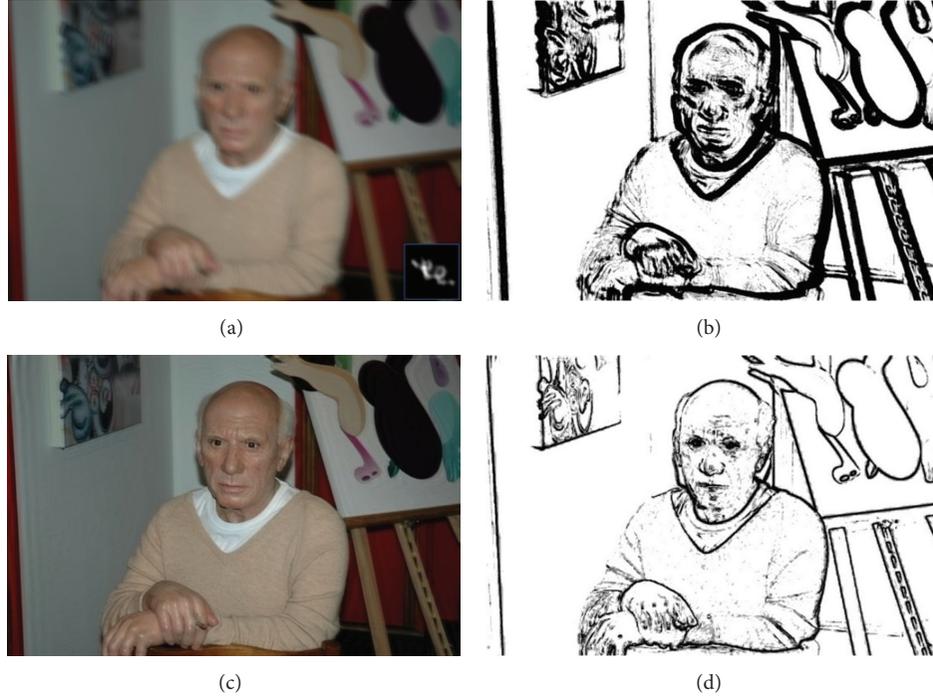


FIGURE 2: Reference map estimation. (a) Input blurred image. (b) First reference map is estimated from the blurred image. It can locally compute smooth regions and distinguish different edge regions. (c) The deconvolved image with the first adaptive regularization. (d) Second reference map is estimated from previous deconvolved result.

result from the previous iteration to distinguish different edge regions elaborately. We also adopt the same criterion to extract map. Second estimated result is represented in Figure 2(d). The map will be renewed from the earlier deconvolved result after each iteration.

3.2. Image Deconvolution with Adaptive Regularization. Now, we would like to seek the MAP solution for the latent image in the deconvolution stage. But the limitation of conventional regularization is that the same value of weighting coefficient η is applied to whole pixels. To overcome this fault and balance the image quality, our method introduces an improvement, that is, to apply appropriate regularization weight according to the local characteristics. So, the adaptive regularization is performed based on the estimated reference map during each iteration time. Based on this idea, we simply modify (6) as follows:

$$\arg \min_I \sum_{i=1}^n \left((B - K \otimes I)_i^2 + \eta_p \sum_{k=1}^2 |(f_k \otimes I)_i|^q \right), \quad (10)$$

where the weighting coefficient η_p is adjusted based on the estimated reference map. It provides a basis to adaptively suppress the ringing effects in different regions. The effect of adaptive regularization in deconvolution result is represented in Figure 3. In comparison, our result exhibits sharper image detail and fewer artifacts.

However, the use of sparse distributions with $q < 1$ makes the optimization problem nonconvex. It becomes slow

to solve the approximation. Using the half-quadratic splitting, Krishnan's [7] fast algorithm introduces two auxiliary variables ω_1 and ω_2 at each pixel to move the $(f_k \otimes I)_i$ terms outside the $|\cdot|^q$ expression. Thus, (10) can be converted to the following optimization problem:

$$\arg \min_{I, \omega} \sum_{i=1}^n \left((B - K \otimes I)_i^2 + \frac{\beta}{2} ((f_1 \otimes I - \omega_1)_i^2 + (f_2 \otimes I - \omega_2)_i^2) + \eta_p (|(\omega_1)_i|^q + |(\omega_2)_i|^q) \right), \quad (11)$$

where $(f_k \otimes I - \omega_k)^2$ term is for constraint of $f_k \otimes I = \omega_k$ and β is a control parameter that we will vary during the iteration process. As β parameter becomes large, the solution of (13) converges into that of (12). This scheme is also called alternating minimization [12] where we adopt a common technique for image restoration. Minimizing (13) for a fixed β can be performed by alternating two steps. This means that we solve ω and I , respectively.

One subproblem is to solve ω_1 and ω_2 , which is called ω subproblem. First, the initial I is set to the input blurred image B . Given a fixed I , finding the optimal ω can be simplified into the following optimization problem:

$$\arg \min_{\omega} \left(\frac{\lambda_p}{2} |\omega|^q + \frac{\beta}{2} (\omega - \nu)^2 \right), \quad (12)$$

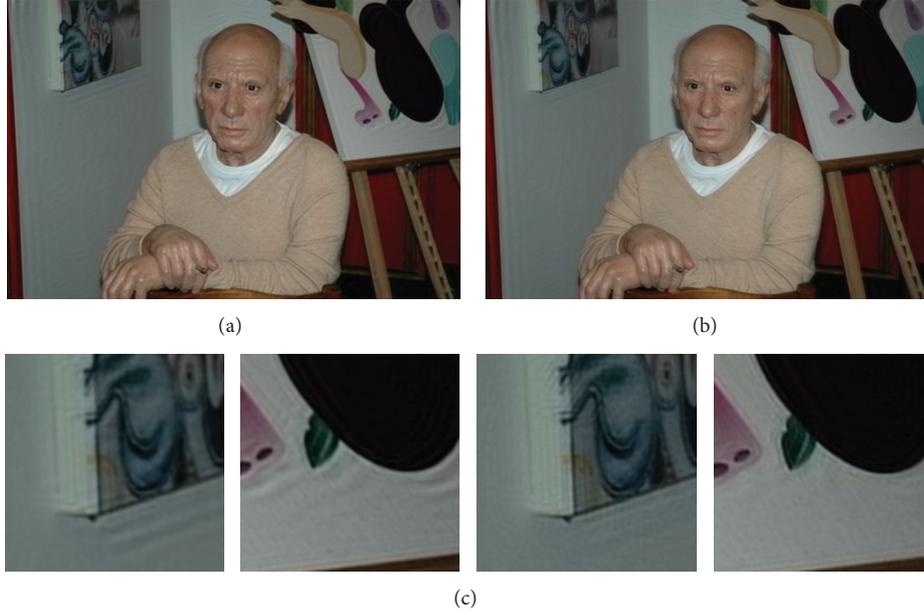


FIGURE 3: Effect of deconvolution with adaptive regularization. (a) The deconvolved result by Krishnan's [7] algorithm with the same weighting value. (b) Our deconvolved result with different η_p weighting values according to the local characteristics as defined above. (c) Close-ups of two image regions for comparison.

where the value $\nu = (f_k \otimes I)$. For convenient calculation, η_p is replaced with $\lambda_p/2$, and it does not affect the performance. In particular for the case $q = 2/3$, the ω satisfying the above equation is the analytical solution of the following quartic polynomial:

$$\omega^4 - 3\nu\omega^3 + 3\nu^2\omega^2 - \nu^3\omega + \frac{\lambda_p^3}{27\beta^3} = 0; \quad (13)$$

to find and select the correct roots of the above quartic polynomial, we adopt Krishnan's approach, as detailed in [7]. They numerically solve (12) for all pixels using a lookup table to find ω .

Second, we briefly describe the I subproblem and its straightforward solution. Given a fixed value of ω from previous iteration, we aim to obtain the optimal I by the

following optimization problem. Equation (11) is modified as

$$\arg \min_I \left((B - K \otimes I)^2 + \frac{\beta}{2} \left((f_1 \otimes I - \omega_1)^2 + (f_2 \otimes I - \omega_2)^2 \right) \right); \quad (14)$$

the I subproblem can be optimized by setting the derivative of the cost function to zero. So, (14) is quadratic in I . The optimal I is

$$\left(F_1^T F_1 + F_2^T F_2 + \frac{2}{\beta} K^T K \right) I = F_1^T \omega_1 + F_2^T \omega_2 + \frac{2}{\beta} K^T B; \quad (15)$$

for brevity, we set $F_k I = (f_k \otimes I)$ and $KI = (K \otimes I)$. Assuming circular boundary conditions, (15) can apply 2D FFTs which diagonalize the convolution matrices F_1 , F_2 , and K , helping us to find optimal I directly:

$$I = \text{IFFT} \left(\frac{\text{FFT}(F_1)^* \circ \text{FFT}(\omega_1) + \text{FFT}(F_2)^* \circ \text{FFT}(\omega_2) + (2/\beta) \text{FFT}(K)^* \circ \text{FFT}(B)}{\text{FFT}(F_1)^* \circ \text{FFT}(F_1) + \text{FFT}(F_2)^* \circ \text{FFT}(F_2) + (2/\beta) \text{FFT}(K)^* \circ \text{FFT}(K)} \right), \quad (16)$$

where $*$ denotes the complex conjugate and \circ denotes component-wise multiplication. The $\text{FFT}(\cdot)$ and $\text{IFFT}(\cdot)$ represent Fourier and inverse Fourier transforms respectively. Solving (16) requires only three FFTs at each iteration since some FFTs operation of F_1 , F_2 , and K can be precomputed.

The nonconvex optimization problem arises from the use of a hyper-Laplacian prior with $q < 1$, then we

adopt a splitting approach that allows the nonconvexity to become separable over pixels. We now give the summary of alternating minimization scheme. As described above, we minimize (11) by alternating the ω and I subproblems, before increasing the value of β and repeating. At the beginning of each iteration, we first compute ω given the initial or deconvolved I by minimizing (14). Then, given a fixed value

of ω , it can minimize (14) to find the optimal I . The iteration process stops when the parameter β is sufficiently large. In practice, starting with small value β we scale it by an integer power until it exceeds some fixed values β_{Max} .

Finally, we mention one last improvement detail. In the blurring process, the convolution operator makes use of not only the image inside the field of view (FOV) of the given observation but also part of the scene in the area bordering it. The part outside the FOV cannot be available to the deconvolution process. When any algorithm performs deconvolution in the Fourier domain, this missing information would cause artifacts at the image boundaries. To analyze this cause, the FFTs assume the periodicity of the data, the missing pixels will be taken from the opposite side of the image when FFTs are performed, since the data obtained may not coincide with the missing ground truth data. These give rise to boundary artifacts, which poses a difficulty in various restoration methods.

To reduce the visual artifacts caused by the boundary value problem, we use Liu's approach, as detailed in [13]. The basic concept to solve the problem is to expand the blurred image such that the intensity and gradient are maintained at the border between the input image and expanded block. This algorithm suppresses the ripples near the image border. It does not require the extra assumption of the PSF and can be adopted by any FFTs based restoration methods.

4. Experimental Results

4.1. Parameters Setting. Since we want to suppress the contrast of ringing in the extracted smooth regions while avoiding suppression of sharp edges, the regularization strength should be large in smooth regions and small in others. We also concern the main edges of the textured regions and other smaller or fainter features. So we controlled regularization strength referring to the different edge property. This means that we need the most weak regularization for the main edges regions.

Through all the experiments, the weighting coefficient η_p is controlled adaptively referring to estimated maps. We have used a geometric progression for the different regions of values of $\eta_{p'} = \eta_p/r$. The rate r is set to 3; that is, we decrease the regularization strength according to the local characteristics. In our experiments, the η_p is set to 1×10^{-3} in smooth regions. Then we decrease the value to 3.3×10^{-4} in next smaller or fainter edges, 1.1×10^{-4} in the strong edges, and so on. The main edges will be applied to the least value of η_p , and the $\lambda_p/2$ is equal to η_p in (14). The η_p also progressively decrease as the number of iterations is increased. It can produce the best results. In addition, our method uses a hyper-Laplacian prior with $q = 2/3$. To find the optimal solution, parameter β will vary during the iterative process. The β value is varied from 1 to 256 by integer powers of $2\sqrt{2}$. As β is larger than $\beta_{\text{Max}} = 256$, the iterations will stop.

Besides, for all testing images, in our experiments, we covert the RGB color image to the YCbCr color space and only the luminance channel is taken into computation. Furthermore, with the alternating minimization and FFTs operation, those schemes can accelerate total computational time. In the next two parts we will demonstrate the effectiveness of our approach.

4.2. Synthetic Images. The synthesized degradations are generated by convolving the artificial PSF and the sharp images. In the first part of the experiments, both subjective quality and objective quality are compared. For testing the subjective performance, we compare our result against those of three existing nonblind image deconvolution methods, as shown in Figure 4.

Figure 4 shows the comparison of the visual quality for the simulated image. The standard RL method preserves edges well but produces the severe ringing artifacts. More iteration introduces not only more image details but also more ringing. Besides, the Matlab RL function is performed in the frequency domain, so it gives rise to severe boundary artifacts. Levin's and Krishnan's methods reduce ringing effectively due to advanced image priors. But they suppress or blur some details since regularization strength is applied to all pixels with the same value. The IRLS solution of Levin's algorithm also expands expensive running times. Our approach can recover finer image details and thin image structures while successfully suppressing ringing artifacts.

In addition, for testing the visual objective quality, the peak signal to noise ratio (PSNR) measurement is also used to evaluate quantitatively the quality of above restored result. Given signal I , the PSNR value of its estimate is defined as:

$$\text{PSNR (dB)} = 10 \cdot \log \frac{255^2}{(1/mn) \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I_{ij} - \widehat{I}_{ij})^2}, \quad (17)$$

where $m \times n$ is the size of the input image, I_{ij} is the intensity value of true clear image at the pixel location (i, j) , and \widehat{I}_{ij} corresponds to the intensity value of the restored image at the same location. The average PSNR values of above results are listed in Table 1.

4.3. Real Blurred Images. Besides testing the method on synthetic degradations, we also apply our method to real life blurred degradations. We estimate the PSF of these images by Xu's method [4]. For the real blurred image, only subjective quality is measured, as shown in Figures 5, 6, and 7.

Figures 5 and 6 show more results of the images from [3]. The PSF is estimated by [4]. Then, the inferred PSF is used as the input of all nonblind methods for comparison. With estimated PSF, which is usually not accurate, the deconvolution methods need to perform robustly. The fine details of the thin branch are compared in Figure 5. It can



FIGURE 4: Deblurring results of synthetic photo “Lena.” (a) The size of blurred image is 512×512 , and the estimated PSF size is 33×39 . (b) The standard Richardson-Lucy method. (c) The result of Levin’s method [5] with the sparse prior. (d) The result of Krishnan’s method [7] with the hyper-Laplacian prior. (e) The result of our deconvolution with adaptive regularization. (f) Close-ups of (a)–(e) image regions for comparison.

be verified that our approach shows superior edge preserving ability compared to other nonblind methods. In Figure 6, standard RL produces the most noticeable ringing even preserving the sharp edges. Our restoration result balances the details recovery and ringing suppression. Finally, the degraded image of Figure 7 uses a large aperture and the focused object is blurry due to camera shake. In comparison, our deconvolution result exhibits richer and clearer image structures than other methods.

Finally, the computational costs of our proposed method and other nonblind image deconvolution methods

considered in the experiments are compared in Table 2. All codes are tested in the computer with AMD Phenom II X4 965 3.40 GHz processor and 4.0 GB RAM. The Matlab RL method is performed in the frequency domain with 20 iterations. For the IRLS scheme, we used the implementation of [5] with default parameters. Krishnan’s method and our proposed method both use hyper-Laplacian prior to regularize the solution, and we set the same β value in the experiment. Experimental results show that the proposed adaptive scheme consumes acceptable processing time while preserving more details in the deblurred images.

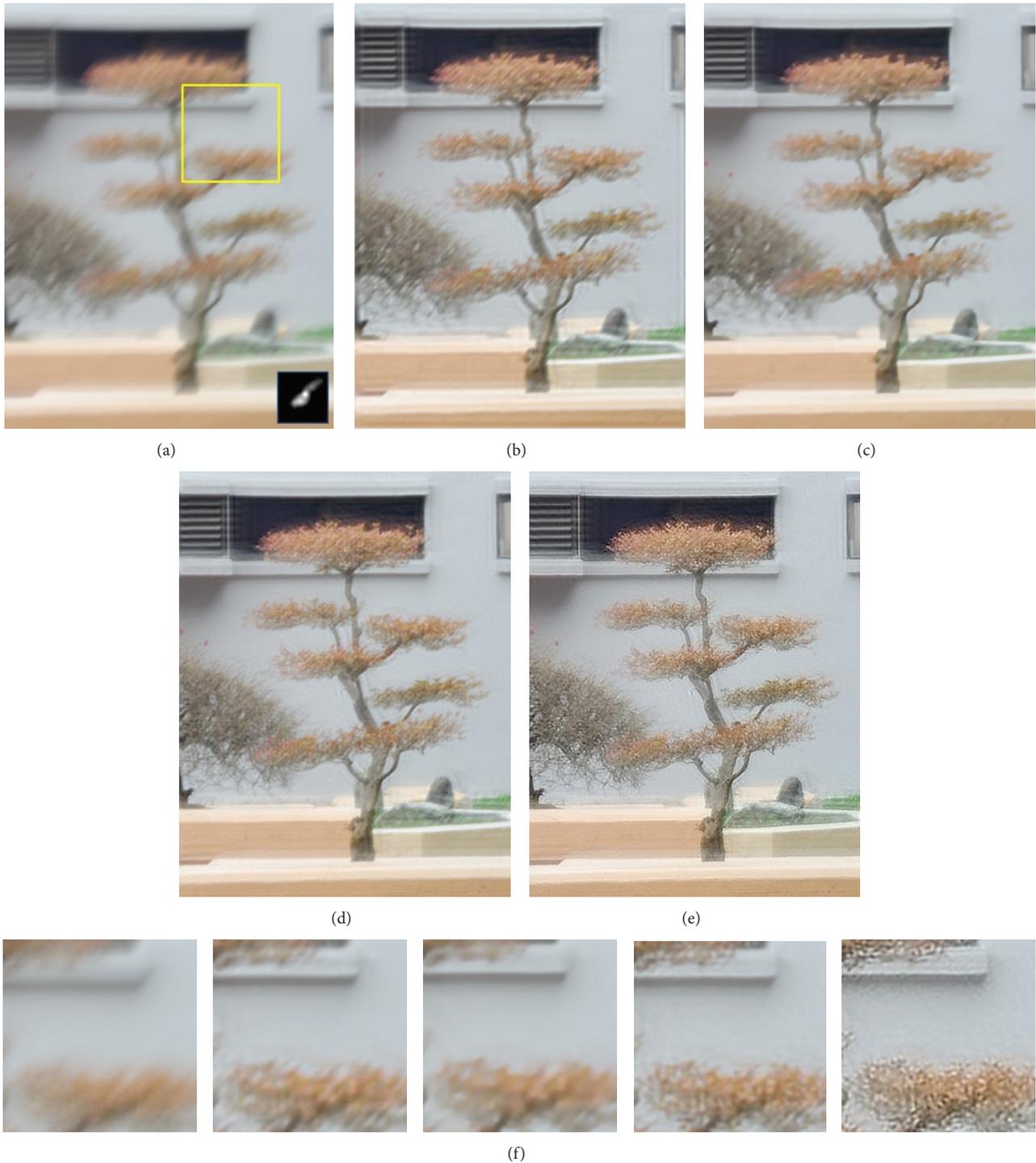


FIGURE 5: Deblurring results of real photo “Red tree.” (a) The size of blurred image is 454×588 , and the estimated PSF size is 27×27 . (b) The standard Richardson-Lucy method. (c) The result of Levin’s method [5] with the sparse prior. (d) The result of Krishnan’s method [7]. (e) The result of our deconvolution with adaptive regularization. (f) Close-ups of (a)–(e) image regions for comparison.

5. Conclusions

The image deblurring is a long-standing problem for many applications. In this paper, a high-quality nonblind deconvolution to remove camera motion blur from a single image

has been presented. We follow the regularization based framework using natural image prior to constrain the optimal solution. Our main contribution is an effective scheme for balancing the image details recovery and ringing suppression. We first introduce the reference map indicating the



FIGURE 6: Deblurring results of real photo "Statue." (a) The size of blurred image is 800×800 , and the estimated PSF size is 31×31 . (b) The standard Richardson-Lucy method. (c) The result of Levin's method [5] with the sparse prior. (d) The result of Krishnan's method [7]. (e) The result of our deconvolution with adaptive regularization. (f) Close-ups of (a)–(e) image regions for comparison.

TABLE 1: Comparison of average PSNRs (dB).

Image	Blurry	Richardson-Lucy [16, 17]	Levin et al., 2007 [5]	Krishnan and Fergus, 2009 [7]	Ours
Lena	23.99	29.69	31.06	31.23	31.76
Fruits	23.51	28.11	29.03	30.29	30.71
Airplane	23.22	28.50	33.45	33.39	34.02

smooth regions and different textured edge regions. Then, according to the image local characteristics, we can control regularization weighting factor adaptively. In addition, the proposed method is practically considering the complexity by FFTs operations. Deconvolved results obtained by our

approach show a noticeable improvement in recovering visually pleasing details with fewer ringing.

In the future, the deblurring topic is still more challenging and exciting. We have found that the severe noise, PSF estimation errors, or large PSF may be propagated and



FIGURE 7: Deblurring results of real photo “Flower.” (a) The size of blurred image is 533×800 , and the estimated PSF size is 31×31 . (b) The standard Richardson-Lucy method. (c) The result of Levin’s method [5] with the sparse prior. (d) The result of Krishnan’s method [7]. (e) The result of our deconvolution with adaptive regularization. (f) Close-ups of (a)–(e) image regions for comparison.

amplified the unpleasant artifacts. Thus, the research direction of future work is how to estimate accurate PSF and propose the robust blind deblurring model. Recently, the spatially variant PSF models have also drawn some attention for better modeling practical motion blurring operator [14, 15]. This means that the PSFs are not uniform in appearance,

for example, from slight camera rotation or nonuniform objects movement. It needs to explore the removal of shift-variant blur using a general kernel assumption. Moreover, we are also interested to apply the proposed framework to other restoration problems, such as image denoising, video deblurring, or surface reconstruction.

TABLE 2: Comparison of the computational costs (Seconds).

Image	Richardson-Lucy	Levin et al., 2007 [5]	Krishnan and Fergus, 2009 [7]	Ours
Lena	12.54	323.21	7.82	8.90
Fruits	14.20	354.86	8.03	10.69
Airplane	12.98	328.11	6.31	8.62
Red tree	13.21	672.75	6.68	10.09
Statue	15.22	1060.54	8.93	11.69
Flower	14.27	989.14	8.32	11.12

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank the anonymous reviewers. Their comments helped in improving this paper. This research is supported by the National Science Council, Taiwan, under Grants NSC 101-2221-E-005-088 and NSC 102-2221-E-005-083.

References

- [1] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 787–794, 2006.
- [2] N. Joshi, R. Szeliski, and D. J. Kriegman, "PSF estimation using sharp edge prediction," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [3] S. Cho and S. Lee, "Fast motion deblurring," *ACM Transactions on Graphics*, vol. 28, no. 5, article 145, 2009.
- [4] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proceedings of the 11th European Conference on Computer Vision (ECCV '10)*, pp. 157–170, 2010.
- [5] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Transactions on Graphics*, vol. 26, no. 3, Article ID 1276464, 2007.
- [6] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Progressive inter-scale and intra-scale non-blind image deconvolution," *ACM Transactions on Graphics*, vol. 27, no. 3, article 74, 2008.
- [7] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 1033–1041, December 2009.
- [8] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Transactions on Graphics*, vol. 27, no. 3, article 73, 2008.
- [9] W. Yongpan, F. Huajun, X. Zhihai, L. Qi, and D. Chaoyue, "An improved Richardson-Lucy algorithm based on local prior," *Optics and Laser Technology*, vol. 42, no. 5, pp. 845–849, 2010.
- [10] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman, "A content-aware image prior," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 169–176, San Francisco, Calif, USA, June 2010.
- [11] J.-H. Lee and Y.-S. Ho, "High-quality non-blind image deconvolution with adaptive regularization," *Journal of Visual Communication and Image Representation*, vol. 22, no. 7, pp. 653–663, 2011.
- [12] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 248–272, 2008.
- [13] R. Liu and J. Jia, "Reducing boundary artifacts in image deconvolution," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '08)*, pp. 505–508, San Francisco, Calif, USA, October 2008.
- [14] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding blind deconvolution algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2354–2367, 2011.
- [15] Y.-W. Tai, P. Tan, and M. S. Brown, "Richardson-Lucy deblurring for scenes under a projective motion path," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1603–1618, 2011.
- [16] W. H. Richardson, "Bayesian-based iterative method of image restoration," *Journal of the Optical Society of America*, vol. 62, no. 1, pp. 55–59, 1972.
- [17] L. Lucy, "An iterative technique for rectification of observed distributions," *Astronomical Journal*, vol. 79, no. 6, pp. 745–765, 1974.

Research Article

A Support Vector Data Description Committee for Face Detection

Yi-Hung Liu,¹ Yung Ting,¹ Shian-Shing Shyu,² Chang-Kuo Chen,²
Chung-Lin Lee,² and Mu-Der Jeng³

¹ Department of Mechanical Engineering, Chung Yuan Christian University, Chungli 320, Taiwan

² Institute of Nuclear Energy Research, Atomic Energy Council, Taoyuan 325, Taiwan

³ Department of Electrical Engineering, National Taiwan Ocean University, Keelung 200, Taiwan

Correspondence should be addressed to Yi-Hung Liu; lyh@cycu.edu.tw

Received 4 November 2013; Accepted 24 December 2013; Published 19 February 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Yi-Hung Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Face detection is a crucial prestage for face recognition and is often treated as a binary (face and nonface) classification problem. While this strategy is simple to implement, face detection accuracy would drop when nonface training patterns are undersampled. To avoid these problems, we propose in this paper a one-class learning-based face detector called support vector data description (SVDD) committee, which consists of several SVDD members, each of which is trained on a subset of face patterns. Nonfaces are not required in the training of the SVDD committee. Therefore, the face detection accuracy of SVDD committee is independent of the nonface training patterns. Moreover, the proposed SVDD committee is also able to improve generalization ability of the original SVDD when the face data set has a multicluster distribution. Experiments carried out on the extended MIT face data set show that the proposed SVDD committee can achieve better face detection accuracy than the widely used SVM face detector and performs better than other one-class classifiers, including the original SVDD and the kernel principal component analysis (Kernel PCA).

1. Introduction

Face detection plays a key role in human-computer interaction since it is a prior step to face recognition. Given an image, the objective of face detection is to locate the faces in the image and return the location of each face. Due to complex backgrounds, variations in facial details, and lighting conditions, face detection has been considered one of the most challenging pattern recognition problems. A large body of works has been presented to tackle this difficult problem in the past decades, which had been nicely surveyed in [1–3].

1.1. Related Works. Appearance-based approach has dominated the recent advances in face detection [3], which consists of two main steps: first, a sliding window is used for scanning the whole image in a serial fashion [4, 5]; then a preselected face detector performs a binary (face and nonface) classification task on each window image to verify whether a face is present or not in each window. Previous works

based on this approach were focused on addressing issues such as (1) exploiting robust features, for example, Haar-like features [6], Bayesian feature [7], spectral histogram [8], local binary pattern- (LBP-)based spatial histogram [9], and principal component analysis (PCA) and its nonlinear version: Kernel PCA [10], (2) seeking face detectors with good generalization ability, such as neural networks (NNs) [11–13], Bayesian classifier [7, 14], and support vector machine (SVM) [4, 8, 10, 15–25], and (3) further improving efficiency of a given face detector by boosting-based techniques, in which the AdaBoost is probably the most famous and has been used in the Viola-Jones face detector [6, 26]. In this work, we aim at dealing with the second issue, the face detector design, and propose a novel face detector called support vector data description (SVDD) committee.

1.2. Problem Description. Appearance-based face detection typically treats the face detection task as a binary classification problem [2, 3, 5]: face and nonface classification. Accordingly, two-class classifiers were adopted. According to previous

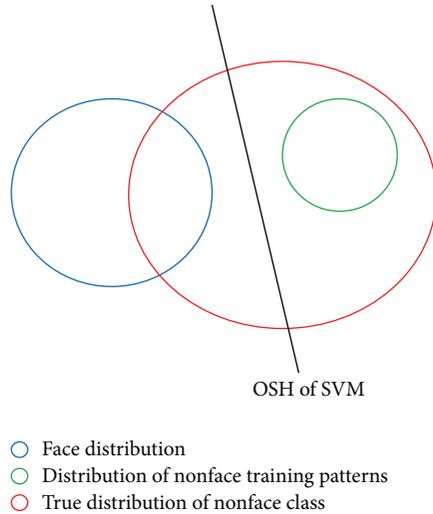


FIGURE 1: This figure shows that if an SVM is trained on a training set in which the nonface training set does not represent the nonface class, some of nonface testing patterns that lie inside of the red boundary would be classified as face patterns (i.e., falling into the wrong side of the SVM's OSH), resulting in a high false positive rate in the testing stage.

works, the two-class classifier SVM by Vapnik [27] has been frequently used as the face detector. The success of SVM in face detection should be attributed to the use of kernel tricks and its learning strategy based on structural risk minimization (SRM). However, SVM may still suffer from a critical problem when applying it to face detection: the high false negative rate due to unrepresentative nonface training patterns, described as follows.

To train an SVM, one has to prepare a training set containing face (positive) and nonface (negative) patterns in prior [22]. The training set is then used to train an SVM to find an optimal separating hyperplane (OSH) with maximal margin of separation in a kernel-induced feature space. However, compared with face class, nonface class's distribution has a relatively large variation due to rich nonface patterns. In other words, it is easy to collect a set of face training patterns which can represent the face class; however, collecting a set of nonface training patterns which are representative enough is difficult because any patterns that do not belong to the face class are nonface patterns. In other words, the nonface class is most likely undersampled: distribution of the collected nonface patterns used for training is not identical to the true distribution of the nonface class. If an SVM is trained by such a training set in which the nonface training set is unrepresentative, many nonface patterns would fall into the wrong side of the OSH in testing stage, resulting in numerous false positives, as illustrated in Figure 1.

1.3. Presented Work. To avoid this critical problem, we choose to adopt the one-class learning strategy to deal with aforementioned undersampled problem of nonface training patterns. One-class learning is to solve the conventional two-class classification problems where one of the two classes is undersampled, or only the data of one single class can

be available for training [28–30]. One-class classifiers are to find a compact description for a specific class (usually being referred to as *target class*) and can be built on just one single class, the target class. In this work, we treat faces as targets, while nonfaces as outliers. The decision boundary of a one-class classifier is then used to distinguish targets from outliers.

Various one-class classifiers have been proposed, such as the linear programming (LP) approach [31], single-class minimax probability machine (MPM) [32], Gaussian mixture model (GMM) [28], one-class SVM [33], SVDD [34, 35], and the Kernel PCA for novelty detection [36]. In this work, we design a robust face detector based on the SVDD of Tax and Duin.

SVDD is a kernel method for novelty detection. Given a target training set, SVDD maps the set into a higher-dimensional kernel-induced feature space and then finds a minimum-volume hypersphere that can enclose all or most of the mapped target data in this feature space. Due to the use of kernel tricks, the sphere boundary in the feature space becomes a flexible one in the original input space, thus being able to fit any irregularly shaped target data sets. This is particularly useful for face detection since face patterns are in general nonlinearly distributed [37, 38]. Recently, success of SVDD has been shown in a variety of novelty detection problems, such as anomaly detection in hyperspectral images [39], defect inspection [40], and novel percept detection for a vision-guidance mobile robot [41].

However, SVDD still has its limits. When a target training set is not a compact set but is formed by a set of disjoint clusters in the data space, the generalization performance of SVDD would drop significantly, as pointed out by Tax and Duin [34]. Unfortunately, face patterns from different individuals form a multicluster distribution in the space of patterns. Thus, using one single SVDD to discriminate faces from nonfaces may not be adequate. To solve this problem, we propose in this paper an SVDD committee.

The training of the proposed SVDD committee consists of two stages. In the first stage, a given face training set is automatically partitioned into disjoint clusters using fuzzy *c*-means (FCM) algorithm [42] and a partitioning entropy-based best cluster selection criterion [43]. The face patterns in each cluster form a compact face subset. Each face subset is then used to train a unique SVDD, which is the second stage. In addition, the decision boundary of SVDD often encloses the face training patterns tightly, which limits the generalization performance for testing faces. To improve the performance, we also modify the original decision function of SVDD so that the decision boundary of SVDD is enlarged. By doing so, face acceptance rate can be improved. Finally, if there are *c* face clusters, *c* SVDDs (members) will be trained. In the testing stage, each trained SVDD serves as a committee member. An unseen pattern is classified as a face pattern if it is accepted by any of the *c* SVDDs. Details of the SVDD will be introduced in Section 3.

The rest of this paper is organized as follows. In Section 2, we first review the basics of SVDD. Then, the SVDD committee will be introduced in Section 3. Results and discussions are provided in Section 4. Finally, conclusions are drawn in Section 5.

2. SVDD

Let $T = \{\mathbf{x}_i \in \mathbf{R}^{d_1 N}\}_{i=1}^N$ be a face training set, where \mathbf{x}_i are face training patterns. SVDD maps the training patterns into a higher dimensional space F using a nonlinear map $\phi : \mathbf{R}^d \rightarrow F$ and then finds a minimum-enclosing hypersphere with center \mathbf{a}_F and radius R in F , which can be formulated as the optimization problem as

$$\begin{aligned} & \text{Minimize} \quad R^2 + C \sum_{i=1}^N \xi_i \\ & \text{subject to} \quad \|\phi(\mathbf{x}_i) - \mathbf{a}_F\|^2 \leq R^2 + \xi_i, \\ & \quad \quad \quad \xi_i \geq 0, \quad \forall i = 1, \dots, N, \end{aligned} \quad (1)$$

where the penalty weight C is user specified and ξ_i are slack variables representing training errors. Taking the partial derivatives $\partial L / \partial R = 0$, $\partial L / \partial \mathbf{a}_F = 0$ and $\partial L / \partial \xi_i = 0$, where

$$L = R^2 + C \sum_{i=1}^N \xi_i - \sum_i \alpha_i (R^2 + \xi_i - \|\phi(\mathbf{x}_i) - \mathbf{a}_F\|^2) - \sum_i \beta_i \xi_i \quad (2)$$

is the Lagrangian function and α_i and β_i are nonnegative Lagrange multipliers, and substituting the results back into L yield the dual constrained optimization problem as follows:

$$\begin{aligned} & \text{Maximize} \quad \sum_{i=1}^N \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & \text{subject to} \quad \sum_{i=1}^N \alpha_i = 1, \\ & \quad \quad \quad 0 \leq \alpha_i \leq C, \quad \forall i = 1, \dots, N, \quad C \in \left[\frac{1}{N}, 1 \right], \end{aligned} \quad (3)$$

where $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ is the kernel function. In this paper, the Gaussian function

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right) \quad (4)$$

is used as the SVDD kernel, where σ is the parameter of the Gaussian kernel. According to the Kuhn-Tucker conditions, (1) the data points with $\alpha_i = 0$ are inside of the hypersphere, (2) the data points whose $0 < \alpha_i < C$ are on the sphere boundary, and (3) the data points whose $\alpha_i = C$ fall outside the sphere and have nonzero ξ_i . The data points with $\alpha_i > 0$ are support vectors (SVs). Further, the SVs with $0 < \alpha_i < C$ are called unbounded SVs (UBSVs), while the SVs with $\alpha_i = C$ are called bounded SVs (BSVs). The center of the sphere is spanned by the mapped SVs,

$$\mathbf{a}_F = \sum_{i=1}^{N_S} \alpha_i \phi(\mathbf{x}_i), \quad (5)$$

where N_S is the number of SVs. The sphere radius R is determined by taking any $\mathbf{x}_k \in$ UBSVs and calculating the distance from its image to the center \mathbf{a}_F as follows:

$$\begin{aligned} R &= \sqrt{\|\phi(\mathbf{x}_k) - \mathbf{a}_F\|^2} \\ &= \sqrt{1 - 2 \sum_{\mathbf{x}_i \in \text{SVs}} \alpha_i K(\mathbf{x}_i, \mathbf{x}_k) + \sum_{\mathbf{x}_i \in \text{SVs}} \sum_{\mathbf{x}_j \in \text{SVs}} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)}. \end{aligned} \quad (6)$$

For Gaussian kernel, $K(\mathbf{x}, \mathbf{x}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}) = 1$, for all $\mathbf{x} \in \mathbf{R}^d$. Mapping the sphere boundary $f(\mathbf{x}) = 0$ back into the original space \mathbf{R}^d yields a flexible boundary that encloses the face training set. The free parameter σ controls the tightness of the boundary: the smaller the σ is, the tighter the boundary is. However, the σ cannot be too small; otherwise the boundary will be too tight to get a satisfactory face acceptance rate for unseen patterns. The decision function of SVDD is given by

$$\begin{aligned} f(\mathbf{x}) &= \|\phi(\mathbf{x}) - \mathbf{a}_F\|^2 - R^2 \\ &= (1 - R^2) - 2 \sum_{i=1}^{N_S} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{i,j=1}^N \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j). \end{aligned} \quad (7)$$

If $f(\mathbf{x}) \leq 0$, the test pattern \mathbf{x} is accepted as a face pattern, rejected as a nonface pattern otherwise.

3. SVDD Committee

3.1. Training

Stage 1 (partitioning). The first stage is to partition the face training set $T = \{\mathbf{x}_i \in \mathbf{R}^{d_1 N}\}_{i=1}^N$ into clusters which are disjoint. Here, the fuzzy c -means (FCM) algorithm is employed to accomplish this task, which solves the optimization problem as follows:

$$\text{minimize} \quad J(T; \mathbf{U}, \mathbf{V}) = \sum_{j=1}^c \sum_{i=1}^N u_{ji}^m \|\mathbf{x}_i - \mathbf{v}_j\|^2 \quad (8)$$

subject to the constraints

$$\begin{aligned} & \sum_{j=1}^c u_{ji} = 1, \quad \forall i = 1, \dots, N, \\ & 0 < \sum_{i=1}^N u_{ji} < N, \quad \forall j = 1, \dots, c, \end{aligned} \quad (9)$$

where $\mathbf{U} = [u_{ji}]$ is the partition matrix, $u_{ji} \in [0, 1]$ is the membership degree that the i th training pattern \mathbf{x}_i belongs to the j th cluster, $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_j, \dots, \mathbf{v}_c)$ is a c -tuple of cluster prototypes, $\mathbf{v}_j \in \mathbf{R}^d$ is the centroid of the j th cluster, c is the number of clusters ($2 \leq c \leq N$), and $m \in (1, \infty)$ is the weight controlling the degree of fuzziness of the matrix \mathbf{U} (we set $m = 2$ in this study). FCM algorithm is performed in an iterative manner as follows.

Step 1. Initialize c .

Step 2. Initialize \mathbf{v}_j .

Step 3. Compute u_{ji} by

$$u_{ji} = \frac{\left(1/\|\mathbf{x}_i - \mathbf{v}_j\|^2\right)^{1/(m-1)}}{\sum_{k=1}^c \left(1/\|\mathbf{x}_i - \mathbf{v}_k\|^2\right)^{1/(m-1)}}. \quad (10)$$

Step 4. Update \mathbf{v}_j by

$$\mathbf{v}_j = \frac{\sum_{i=1}^N u_{ji}^m \mathbf{x}_i}{\sum_{i=1}^N u_{ji}^m}, \quad j = 1, \dots, c. \quad (11)$$

Step 5. Repeat Steps 3 and 4 until $\|\mathbf{U}\| < \varepsilon$, where ε is a small positive value.

Choosing a proper number c of clusters is of primary importance. The best number c_{best} of clusters can be determined using the partitioning entropy-based criterion [43] as follows:

$$c_{\text{best}} = \arg \min_{c=2}^N \left\{ \min_{\mathbf{U} \in \Omega_c} [H(\mathbf{U}, c)] \right\}, \quad (12)$$

where Ω_c is the set of solutions and

$$H(\mathbf{U}, c) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^c |u_{ji} \ln u_{ji}| \quad (13)$$

is the partitioning entropy, providing a global validity measure for the clustering results. Finally, the training set T is partitioned into c_{best} subsets using the simple rule: the i th training data point \mathbf{x}_i is assigned to the k th cluster if its membership degree to the k th cluster is the highest; namely,

$$k = \arg \max_{j=1}^{c_{\text{best}}} u_{ji}. \quad (14)$$

Stage 2 (training SVDDs). After the training set T is partitioned into the c_{best} subsets T_i , $i = 1, \dots, c_{\text{best}}$, where there is no overlap between two subsets and $|T| = \sum_{i=1}^{c_{\text{best}}} |T_i|$, the second training stage is to train c_{best} SVDDs. The i th subset is used to train the i th SVDD. Namely, use the face training subset $T_i = \{\mathbf{x}_j \in \mathbf{R}^d\}_{j=1}^{|T_i|}$ to obtain the Lagrange multipliers for the i th SVDD by solving the quadratic programming problem formulated in (3) and then use (6) to calculate the sphere radius R_i . However, the trained decision boundary tightly encloses the face training subset. For some test face patterns located around the subset's distribution, they may be outside of the boundary. Such face test patterns will be rejected as nonfaces as a result. In this study, we deal with this problem by enlarging the sphere; that is,

$$R_i \leftarrow R_i + \Delta R_i, \quad (15)$$

where ΔR_i is a positive value. ΔR_i cannot be too large, otherwise the outlier acceptance rate (nonfaces classified as faces) will be significantly increased although the face

acceptance rate is improved. Therefore, ΔR_i should be much smaller than R_i . According to our preliminary testing results, setting $\Delta R_i = R_i/15$ can improve the face acceptance rate without increasing the nonface acceptance rate. After updating the sphere radius by (15), the decision function $f_i(\mathbf{x})$ for the i th SVDD is obtained.

3.2. Testing. For an unseen pattern \mathbf{x} , it is rejected as a nonface if it is rejected by all the committee members, accepted as a face otherwise:

$$\begin{aligned} \mathbf{x} \in \text{nonface}, & \quad \text{if } f_i(\mathbf{x}) > 0, \quad \forall i = 1, \dots, c_{\text{best}} \\ \mathbf{x} \in \text{face}, & \quad \text{otherwise.} \end{aligned} \quad (16)$$

In other words, the decision making strategy used in the SVDD committee is not based on the usual majority voting. Instead, the unseen pattern is classified as a face as long as at least one of the SVDDs accepts it.

4. Results and Discussions

In Section 4.1, we first introduce the data set used for experiments. Then, in Section 4.2 we give an illustrated example that shows how the proposed SVDD committee deals with the problem of multicluster face distribution by a visualization analysis. Then, we compare our method with SVM and other one-class classifiers in terms of face detection accuracy.

4.1. Data Set. The extended MIT face data set [44, 45] consists of a training set and a test set. The training set contains 489410 patterns, where 17496 are faces and the remaining 471914 are nonfaces. The test set contains 472 faces and 23573 nonfaces. Each pattern is represented by a 361-dimensional vector.

4.2. An Illustrative Example. We randomly select 100 face patterns from the extended MIT face data set and perform the principal component analysis (PCA) to reduce the dimensionality of the patterns by selecting two leading eigenvectors of the pattern covariance matrix. The projections of the 100 face patterns in the 2D PCA-based subspace are depicted in Figure 2.

It can be observed from Figure 2(a) that the face patterns do not form a compact distribution but a multimodal (multicluster) distribution in the 2D space. We first use one single SVDD to learn the boundary which encloses most of the face patterns. As the kernel parameter σ is set to a very large value ($\sigma = 100$), the decision boundary of SVDD is nearly spherical shaped in the original 2D space. As the value of σ decreases to 10 from 100, the SVDD boundary becomes tighter (see Figure 2(c)). However, the boundary is still not tight enough because there still exist empty areas (the areas within the green circles). Nonface patterns falling into these empty areas will be accepted as faces since the patterns are also inside of the SVDD boundary. One way to avoid such a situation is to further decrease the value of σ . However, according to author's previous work [46], when σ is too small, all the mapped target training data will be orthonormal to each other in the Gaussian kernel-induced feature space F

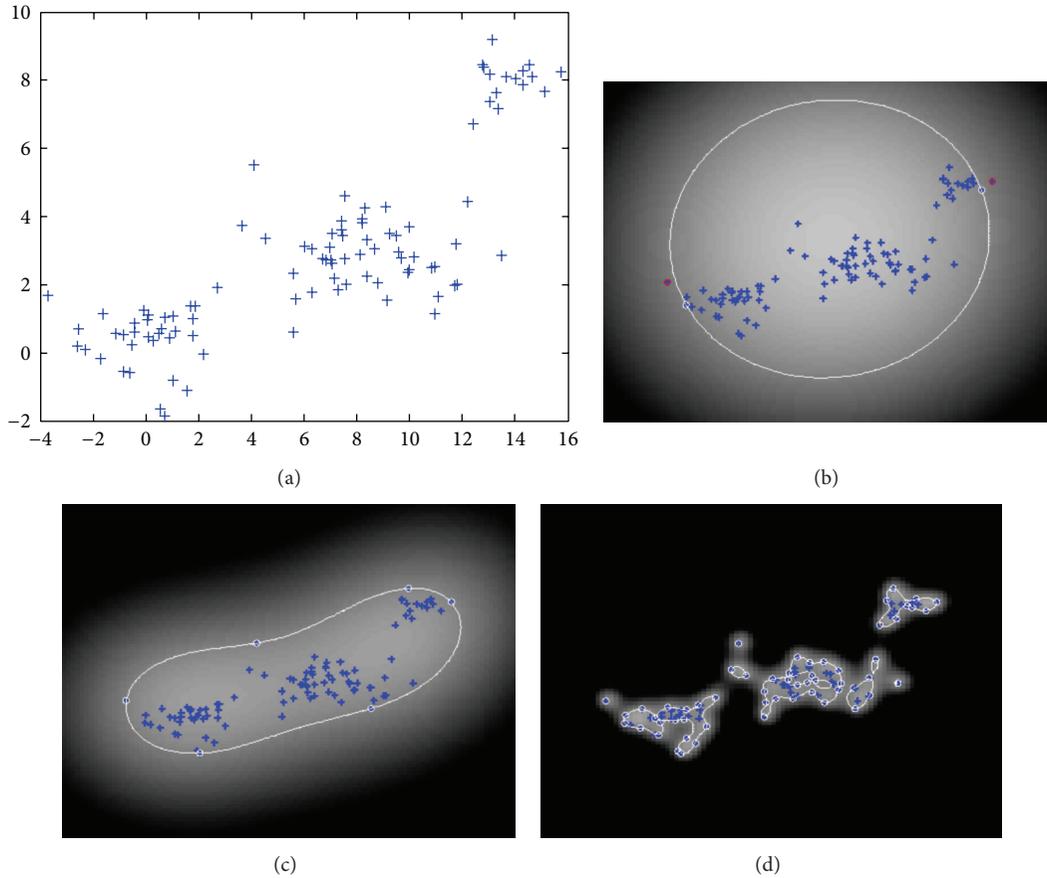


FIGURE 2: (a) Shows the distribution of the randomly selected face patterns in a 2D PCA-based subspace. (b), (c), and (d) show the SVDD training results using $\sigma = 100$, $\sigma = 10$, and $\sigma = 1$, respectively. The white curves are the SVDD decision boundaries. The blue crosses with white circles denote the UBSVs, while the blue crosses with red circles in the upper-right subfigure are BSVs. The penalty weight C is set as 0.3 in all the experiments.

and become SVs. Also, when all or almost all the target training data becomes SVs, the boundary will be too tight to get a good target acceptance rate [33, 34]. As shown in Figure 2(d), the boundary overfits the face training set and most of the face training patterns become SVs as σ is set to 1. Although nonfaces can be rejected by this extremely tight boundary successfully, face test patterns are easily rejected as nonfaces as well, resulting in a poor face acceptance rate (i.e., face detection rate).

Instead of using the whole face training set to train an SVDD, the proposed SVDD committee partitions the whole training set into disjoint clusters and then utilizes the face patterns in each cluster to train an independent SVDD. By doing so, the performance drop due to the multicenter face distribution can thus be solved. An illustrated example is shown in Figure 3.

First, the 100 face patterns are partitioned into three clusters (three disjoint subsets) using the FCM algorithm and the best cluster selection criterion stated in Section 3. Compared with the whole training set, each subset forms a much more compact distribution. Then, each subset is utilized to train an SVDD. Finally, the three independent SVDDs constitute a committee. By comparing the results of

Figure 3 with the ones of Figure 2, we can see that using multiple SVDDs to find a description for a multicenter face distribution is more suitable than using one single SVDD.

4.3. Comparison with Other Methods

4.3.1. Methods. We compare our proposed SVDD committee with the frequently adopted face detector SVM and the one-class learning methods including the regular SVDD (i.e., single SVDD) and Kernel PCA [36].

Kernel PCA is a nonlinear version of PCA, which was originally designed for pattern representation [47]. Recently, Tsang et al. [44] further extended their original idea to novelty detection. Kernel PCA uses the reconstruction error in a kernel-induced feature space as a novelty measure (see Section 3 of [44] for details). A test data point is accepted as a target if its reconstruction error is below a predefined threshold, rejected as an outlier otherwise. The Kernel PCA for novelty detection involves three free parameters: kernel parameter, number of chosen eigenvectors q , and the threshold of the reconstruction error T_{re} . Free parameters of the methods are listed in Table 1.

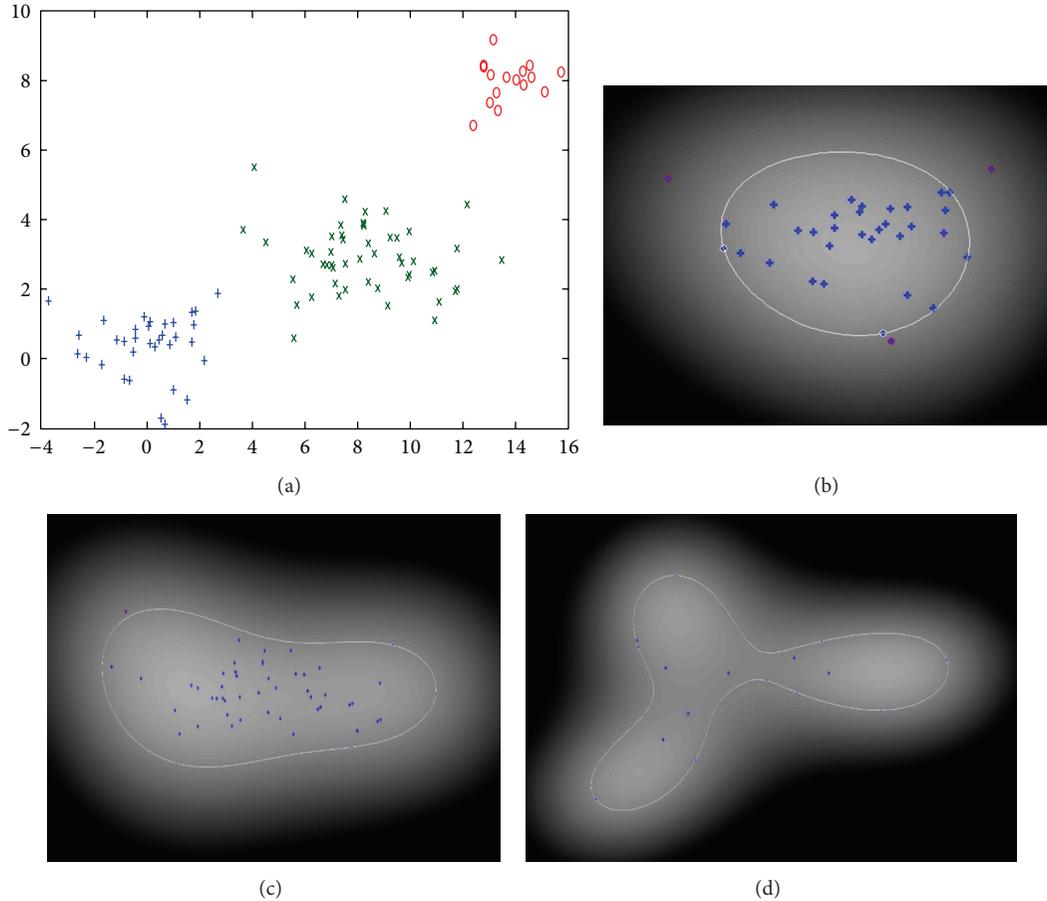


FIGURE 3: (a) shows that the 100 face patterns in the 2D PCA-based subspace are partitioned into three clusters. Face data sets in the three clusters are used to train three independent SVDDs, and the results are displayed in (b) (SVDD for cluster 1), (c) (SVDD for cluster 2), and (d) (SVDD for cluster 3) subfigures, respectively.

TABLE 1: Free parameters of the methods to be compared.

Methods	Parameters
SVM	C, σ
Kernel PCA	σ, q, T_{re}
SVDD	C, σ
SVDD committee	C, σ

4.3.2. *Performance Measure.* For SVM, faces are treated as positive (target) data while nonfaces as negative data (outlier). For one-class classifiers (SVDD, SVDD committee, and Kernel PCA) faces are treated as targets while nonfaces as outliers. For face detection problems, nonfaces often outnumber faces largely. If the data set is imbalanced, usual classification error rate is not an appropriate performance measure [48]. Therefore, the balanced loss

$$l_{bal} = 1 - \frac{TAR + ORR}{2} \quad (17)$$

suggested in [48] is adopted as the performance measure in the following experiments, where TAR and ORR stand for the target acceptance rate and outlier rejection rate, respectively.

A good face detector should be able to achieve a high TAR and a high ORR simultaneously. Therefore, the lower the balanced loss is, the better the face detector is. To facilitate the comparisons, the balanced loss is simply called error rate hereafter.

4.3.3. *Experiments and Results.* All methods are optimized by performing a fivefold cross validation procedure on a data set. The data set is not the entire training set of the extended MIT face data set because the scale of the training set is very large: the training set contains 17496 faces and 471914 nonfaces. Since the intent of this paper is not on proposing a method that can deal with large-scale data sets but on proposing a robust face detector which can be better than the widely used face detector SVM and further improving the performance of the one-class classifier SVDD. Thus, only a subset of the training set is used for training, and a subset of the test set is used for testing. The experiment is as follows.

(1) *Step 1.* Select n_f face and n_{non} nonface patterns from the training set by random. The collected $n_f + n_{non}$ patterns form a new training set. In addition, randomly select 200 face and 400 nonface patterns from the test set. The 600 patterns form

TABLE 2: Comparison of average error rate among different methods on extended MIT face data set (in %).

	Exp A	Exp B	Exp C
SVM	20.12	18.24	14.24
Kernel PCA	16.80	16.03	13.45
SVDD	18.83	17.12	14.27
SVDD committee	14.24	14.02	12.23

a new test set. There is no overlap between the new training set and the new test set.

(2) *Step 2.* Perform 10-run twofold cross validation [30] on the new training set to optimize the methods.

(3) *Step 3.* Feed the new test set to the methods to obtain their error rates.

(4) *Step 4.* Repeat Step 1–Step 3 ten times. Compute the average error rate of each method.

We set $n_f = 500$ and $n_{\text{non}} = 500, 1000, \text{ and } 1500$. There are three experiments in total: Exp A ($n_f = 500, n_{\text{non}} = 1000$), Exp B ($n_f = 500, n_{\text{non}} = 1000$), and Exp C ($n_f = 500, n_{\text{non}} = 1500$). The average error rates of the methods are listed in Table 2.

According to Table 2, we can first observe that the average error rates of SVM in Exp A, Exp B, and Exp C are 20.12%, 18.24%, and 14.24%, respectively. The comparison results indicate that increasing the number of nonface training patterns can improve generalization performance of SVM in face detection. This could be due to the fact that when the number of nonface training patterns increases, the distribution of nonface training patterns can better represent the true distribution of the nonface class, thus being able to gain better generalization performance. Moreover, all the one-class classifiers perform better than the two-class classifier SVM in all experiments except for Exp C where SVM (14.24%) is slightly better than SVDD (14.27%). Also, the proposed SVDD committee gives the best results in the three experiments and greatly improves the accuracy of the original SVDD in face detection, which can be seen from Table 3 where the error reduction ratio (ERR) is defined as

$$\text{ERR} = \frac{e_{\text{SVDD}} - e_{\text{SVDD.Com}}}{e_{\text{SVDD}}}, \quad (18)$$

where e_{SVDD} and $e_{\text{SVDD.Com}}$ denote the average error rates of SVDD and SVDD committee, respectively. The results reported in Table 3 demonstrate that the proposed SVDD committee performs better than one single SVDD in face detection.

4.3.4. Training and Testing Speeds. We further compare the computational complexities of the methods. The training time and testing time of each method are recorded during the experiments. A 3.40 GHz-CPU (i7-3770) computer (with 8 GB RAM) running on Windows 8 is used. The methods are implemented with MATLAB 7.10, which is also a 64-bit

TABLE 3: Error reduction ratios (ERRs) of SVDD committee to SVDD in the three experiments (in %).

	Exp A	Exp B	Exp C
ERR	24.38	18.11	14.30

TABLE 4: Training time of each method (in s).

	Exp A	Exp B	Exp C
SVM	14.56	39.80	120.33
Kernel PCA	2.18	2.11	2.31
SVDD	1.67	1.58	1.66
SVDD committee	3.07	3.23	3.12

TABLE 5: Testing speed of each method (ms/pattern).

	Exp A	Exp B	Exp C
SVM	1.32	2.43	2.69
Kernel PCA	1.31	1.01	1.03
SVDD	0.64	0.45	0.56
SVDD committee	1.31	1.12	1.54

version. Finally, training time and testing are summarized in Tables 4 and 5, respectively.

We can see from Table 4 that the training time of SVM increases with the size of the training set, because SVM' training has $O(n^3)$ time complexity, where $n = n_f/2 + n_{\text{nf}}/2$ (for twofold cross validation, 50% of the patterns are used for training). For one-class classifiers, only half of the face training patterns are included in the training. For example, SVDD's training time complexity is $O(n^3)$, where $n = n_f/2$. Since $n_f = 500$ in all the three experiments, the actual training time of SVDD is almost the same for the three cases. SVDD committee takes more training time than SVDD. However, compared with SVM, SVDD committee has a much faster training speed, especially when the number of training faces increases. The testing speeds reported in Table 5 indicate that SVDD has the fastest testing speed. Although SVDD committee is lower than SVDD, its testing speed (1.12–1.54 ms/pattern) is acceptable for real-time face detection.

5. Conclusion

In this paper, we have presented a novel face detector called SVDD committee. The proposed SVDD committee is based on one-class learning and partitioning strategies, thus being able to improve the generalization performance of the original SVDD in face detection. Also, nonface patterns are not required in the training of SVDD committee. Therefore, the face detection accuracy of SVDD committee would not be affected by the chosen nonface patterns. On the contrary, the frequently adopted face detector SVM is a two-class classifier. Its face detection accuracy depends on the number n_f of collected nonface training patterns, and its training time increases with n_f . Experiments have demonstrated that the proposed SVDD committee not only performs better

than SVM, but also significantly improves the generalization performance of the original SVDD in face detection. Also, the testing speed is acceptable for applications where real-time face detection is required.

This work does not apply any robust feature extraction methods, because the focus of this work is on the development of a novel face detector based on one-class learning. It is believed that the SVDD committee-based face detection accuracy can be further improved if advanced feature extractions are applied, such as LBP and Haar-like features. In addition, the presented work does not address other critical issues such as learning from large-scale data sets. Those will be our future works.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This study was supported by a grant from the Institute of Nuclear Energy Research, Atomic Energy Council, Taiwan.

References

- [1] E. Hjelmås and B. K. Low, "Face detection: a survey," *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236–274, 2001.
- [2] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.
- [3] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," Tech. Rep. MSR-TR-2010-66, Microsoft Corporation, 2010.
- [4] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.
- [5] G. Gualdi, A. Prati, and R. Cucchiara, "Multistage particle windows for fast and accurate object detection," *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 1589–1604, 2012.
- [6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, pp. 1511–1518, December 2001.
- [7] C. Liu, "A Bayesian discriminating features method for face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 725–740, 2003.
- [8] C. A. Waring and X. Liu, "Face detection using spectral histograms and SVMs," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 35, no. 3, pp. 467–476, 2005.
- [9] H. Zhang, W. Gao, X. Chen, and D. Zhao, "Object detection using spatial histogram features," *Image and Vision Computing*, vol. 24, no. 4, pp. 327–341, 2006.
- [10] Y. H. Liu and Y. T. Chen, "Face detection using kernel PCA and imbalanced SVM," in *Advances in Natural Computation*, vol. 4221 of *Lecture Notes in Computer Science*, pp. 351–360, Springer, Berlin, Germany, 2006.
- [11] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.
- [12] R. Féraud, O. J. Bernier, J.-E. Viallet, and M. Collobert, "A fast and accurate face detector based on neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 42–53, 2001.
- [13] C. Garcia and M. Delakis, "Convolutional face finder: a neural architecture for fast and robust face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1408–1423, 2004.
- [14] H. Schneiderman and T. Kanade, "Statistical method for 3D object detection applied to faces and cars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, pp. 746–751, June 2000.
- [15] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 130–136, June 1997.
- [16] T. J. Terrillon, M. N. Shirazi, M. Sadek, H. Fukamachi, and S. Akamatsu, "Invariant face detection with support vector machines," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, vol. 4, pp. 210–217, 2000.
- [17] S. Romdhani, P. Torr, B. Schölkopf, and A. Blake, "Computationally efficient face detection," in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV'01)*, pp. 695–700, Vancouver, Canada, 2001.
- [18] I. Buciu, C. Kotropoulos, and I. Pitas, "Combining support vector machines for accurate face detection," in *IEEE International Conference on Image Processing (ICIP'01)*, pp. 1054–1057, Thessaloniki, Greece, October 2001.
- [19] B. Heisele, T. Serre, S. Prentice, and T. Poggio, "Hierarchical classification and feature reduction for fast face detection with support vector machines," *Pattern Recognition*, vol. 36, no. 9, pp. 2007–2017, 2003.
- [20] M. Rätsch, S. Romdhani, and T. Vetter, "Efficient face detection by a cascaded support vector machine using haar-like features," in *Pattern Recognition*, vol. 3175 of *Lecture Notes in Computer Science*, pp. 62–70, Springer, Berlin, Germany.
- [21] P. Wang and Q. Ji, "Multi-view face detection under complex scene based on combined SVMs," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, pp. 179–182, August 2004.
- [22] J. Chen, R. Wang, S. Yan, S. Shan, X. Chen, and W. Gao, "Enhancing human face detection by resampling examples through manifolds," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 37, no. 6, pp. 1017–1028, 2007.
- [23] G.-Y. Zhang, L.-Y. Guo, J.-H. Wu, H.-M. Li, and G.-Q. Guo, "Face detection using Gabor wavelet and SVM," in *Proceedings of the International Conference on Computer Application and System Modeling (ICCASM'10)*, vol. 2, pp. 588–590, October 2010.
- [24] C. Kyrkou and T. Theodoridis, "SCoPE: towards a systolic array for SVM object detection," *IEEE Embedded Systems Letters*, vol. 1, no. 2, pp. 46–49, 2009.
- [25] M. Kawulok and J. Szymanek, "Precise multi-level face detector for advanced analysis of facial images," *IET Image Processing*, vol. 6, no. 2, pp. 95–103, 2012.
- [26] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

- [27] V. N. Vapnik, *Statistical Learning Theory*, Adaptive and Learning Systems for Signal Processing, Communications and Control Series, John Wiley & Sons, New York, NY, USA, 1998.
- [28] M. Markou and S. Singh, "Novelty detection: a review, part I: statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003.
- [29] M. Markou and S. Singh, "Novelty detection: a review, part II: neural network based approaches," *Signal Processing*, vol. 83, no. 12, pp. 2499–2521, 2003.
- [30] K. Y. Lee, D.-W. Kim, K. H. Lee, and D. Lee, "Density-induced support vector data description," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 284–289, 2007.
- [31] C. Campbell and K. P. Bennett, "A linear programming approach to novelty detection," in *Advances in Neural Information Processing Systems*, vol. 13, pp. 395–401, MIT Press, Cambridge, Mass, USA, 2001.
- [32] G. R. G. Lanckriet, L. E. Ghaoui, and M. I. Jordan, "Robust novelty detection with single-class MPM," in *Advances in Neural Information Processing Systems*, vol. 15, pp. 929–936, MIT Press, Cambridge, Mass, USA, 2003.
- [33] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.
- [34] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognition Letters*, vol. 20, no. 11–13, pp. 1191–1199, 1999.
- [35] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.
- [36] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.
- [37] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithms," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 117–126, 2003.
- [38] Y.-H. Liu and Y.-T. Chen, "Face recognition using total margin-based adaptive fuzzy support vector machines," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 178–192, 2007.
- [39] S. Khazai, A. Safari, B. Mojaradi, and S. Homayouni, "Improving the SVDD approach to hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 4, pp. 594–598, 2012.
- [40] Y.-H. Liu, Y.-K. Huang, and M.-J. Lee, "Automatic inline defect detection for a thin film transistor-liquid crystal display array process using locally linear embedding and support vector data description," *Measurement Science and Technology*, vol. 19, no. 9, Article ID 095501, 2008.
- [41] X. Wang, X. L. Wang, and D. M. Wilkes, "An automated vision based on-line novel percept detection method for a mobil robot," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1279–1294, 2012.
- [42] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY, USA, 1981.
- [43] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent System*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.
- [44] I. W.-H. Tsang, J. T.-Y. Kwok, and J. M. Zurada, "Generalized core vector machines," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1126–1140, 2006.
- [45] I. W. Tsang, A. Kocsor, and J. T. Kwok, "LibCVM," 2011, <http://www.c2i.ntu.edu.sg/ivor/cvm.html>.
- [46] Y.-H. Liu, Y.-C. Liu, and Y.-J. Chen, "Fast support vector data descriptions for novelty detection," *IEEE Transactions on Neural Networks*, vol. 21, no. 8, pp. 1296–1313, 2010.
- [47] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [48] J. Weston, B. Schölkopf, E. Eskin, C. Leslie, and S. Noble, "Dealing with large diagonals in kernel matrices," in *Principles of Data Mining and Knowledge Discovery*, vol. 243 of *Lecture Notes in Computer Science*, pp. 494–511, Springer, Helsinki, Finland, 2002.

Research Article

Optimal $O(1)$ Bilateral Filter with Arbitrary Spatial and Range Kernels Using Sparse Approximation

Shengdong Pan, Xiangjing An, and Hangen He

College of Mechatronics Engineering and Automation, National University of Defence Technology (NUDT), Changsha, Hunan 410073, China

Correspondence should be addressed to Xiangjing An; anxj@nudt.edu.cn

Received 15 October 2013; Accepted 9 January 2014; Published 18 February 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Shengdong Pan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A number of acceleration schemes for speeding up the time-consuming bilateral filter have been proposed in the literature. Among these techniques, the histogram-based bilateral filter trades the flexibility for achieving $O(1)$ computational complexity using box spatial kernel. A recent study shows that this technique can be leveraged for $O(1)$ bilateral filter with arbitrary spatial and range kernels by linearly combining the results of multiple-box bilateral filters. However, this method requires many box bilateral filters to obtain sufficient accuracy when approximating the bilateral filter with a large spatial kernel. In this paper, we propose approximating arbitrary spatial kernel using a fixed number of boxes. It turns out that the multiple-box spatial kernel can be applied in many $O(1)$ acceleration schemes in addition to the histogram-based one. Experiments on the application to the histogram-based acceleration are presented in this paper. Results show that the proposed method has better accuracy in approximating the bilateral filter with Gaussian spatial kernel, compared with the previous histogram-based methods. Furthermore, the performance of the proposed histogram-based bilateral filter is robust with respect to the parameters of the filter kernel.

1. Introduction

The bilateral filter is proposed by Tomasi and Manduchi in [1]. Before this name, it is called SUSAN filter [2]. The general idea of the bilateral filter stems from the earlier work of the neighborhood filter [3] and the sigma filter [4]. The bilateral filter is a noniterative, nonlinear spatial filter which imposes both the geometric affinity and the intensity similarity when aggregating the contribution of each neighboring pixel within the spatial support. Hence the shape of the filter kernel is data adaptive, which enables the filter to preserve abrupt edges when smoothing out small variations. Such an edge-avoiding smoothing property is later demonstrated to be of much potential use in many applications in computer vision and graphics. It has become a general tool in image processing literature, where it is employed in noise cancelation [1, 5–8], high-dynamic-range tone mapping [9–11], image enhancement [12–14], and many other applications [15–17].

On the flip side of its convenient properties for image- and video-based applications, the direct implementation of the standard bilateral filter requires $O(\sigma_s^2)$ operations per pixel, where σ_s is the radius of the effective support of the spatial kernel. The computational complexity is too intensive for time-critical applications. Consequently, a plenty of studies on its simplification and acceleration can be found in the literature. Durand and Dorsey proposed a piecewise linear bilateral filter in [9]. They quantize the intensity into several segments and perform FFT-based linear filtering for each segment. The final results are pooled using a linear interpolation on these linearly filtered images. The complexity of Durand's piecewise linear bilateral filter is $O(\log \sigma_s)$. Pham and van Vliet [18] decomposed the 2D bilateral convolution using two 1D bilateral convolutions and reduced the complexity to $O(\sigma_s)$. Paris et al. [19] and later Chen et al. [11] generalize the idea of the piecewise linear bilateral filter by Durand et al. to form a 3-D bilateral grid. Then the bilateral filtering can be interpreted as a linear convolution of a vector-valued image. Based on an equipollent subsampling in the augmented data

space, the complexity of this method is $O(1 + |\mathcal{R}|/\sigma_s\sigma_r)$, where $|\mathcal{R}|$ denotes the number of grids of the intensity and σ_r is the bandwidth of the range kernel. Thus the bilateral grid method runs faster with larger spatial kernel. Weiss [20] develops an $O(\log\sigma_s)$ algorithm for local histogram calculation, which is later used to derive an $O(\log\sigma_s)$ bilateral filter with box spatial kernel. Porikli [21] proposes an $O(1)$ box bilateral filter by virtue of the $O(1)$ integral histogram [22]. Porikli [21] and Yang et al. [23] and later Chaudhury et al. [24, 25] suggest the use of some series of the intensity to approximate arbitrary range kernel and employ $O(1)$ algorithms to efficiently compute the spatial filtering for each term of the series.

Based on Porikli's $O(1)$ single-box bilateral filter proposed in [21], Gunturk proposes an $O(1)$ bilateral filter with arbitrary spatial and range kernels by linearly combining multiple single-box bilateral filters [26]. The accuracy can be improved compared with the single-box bilateral filter in approximating the bilateral filter with arbitrary spatial kernel. However, when the radius of the effective spatial support is large, a large number of single-box bilateral filters are required to guarantee the accuracy. Furthermore, no instruction for the selection over all the possible box spatial kernels is presented when the number of single-box bilateral filter is limited. Consequently, Gunturk's method does not produce sufficiently good approximation with only a few additional single-box bilateral filters when the spatial kernel is large. More details about the limitations of Gunturk's method are shown in the following text.

In this paper, we focus our attention on approximating arbitrary spatial kernel with a fixed number of boxes. With this constraint, the MSE-based optimization becomes a well-known sparse representation problem. We then employ the Batch-OMP method published recently in [27] to solve the problem for the radiuses and the coefficients of the boxes. In addition to the application to the histogram-based bilateral filter, the proposed method can be employed in other acceleration schemes by substituting the spatial filtering with a fixed times of box filtering. Hence a number of $O(1)$ bilateral filters are produced. Afterwards, the application to the histogram-based bilateral filter is discussed with some representative experiments. Results show that the proposed histogram-based bilateral filter is robust and sufficiently accurate over a large range of the size of the spatial kernel and outperforms the other congeneric methods in both the accuracy and the robustness.

The rest of this paper is arranged as follows. Brief descriptions of the original bilateral filter and its acceleration schemes are provided in Section 2, where the previous accelerations are categorized into several types and analyzed, respectively. Then in Section 3, the proposed method for approximating arbitrary spatial kernel with a fixed number of boxes is deduced in detail. The application to some popular acceleration schemes is also provided in this section. Afterwards, the application to the histogram-based bilateral filter is tested in Section 4, together with some comparisons with the previous histogram-based methods. Finally, the conclusions are drawn in Section 5, followed by some further discussions.

2. Related Works

The bilateral filter is proposed by Tomasi and Manduchi in [1]. It is a normalized convolution in which the contribution of a neighboring pixel depends on both the geometric distance and the intensity difference with regard to the center pixel. Generally, arbitrary kernels can be applied to both the spatial and range filtering to measure the affinity between one pixel and its neighbors. The discrete form of the bilateral filter with arbitrary spatial and range kernels can be formulated as follows:

$$u^{\text{BF}}(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in N(\mathbf{x})} K_s(\mathbf{x} - \mathbf{y}) K_r(v(\mathbf{x}) - v(\mathbf{y})) v(\mathbf{y})}{\sum_{\mathbf{y} \in N(\mathbf{x})} K_s(\mathbf{x} - \mathbf{y}) K_r(v(\mathbf{x}) - v(\mathbf{y}))}, \quad (1)$$

where v and u are, respectively, the original image and the filtered image and $N(\mathbf{x})$ denotes the effective spatial support of the filtering kernel, which is centered at \mathbf{x} . K_s and K_r are, respectively, the spatial and range kernels. The original bilateral filter employs Gaussian kernels for both the spatial and range filtering, which provides intuitive control over the similarity measure with the variances, respectively, for these two kernels. However, the brute-force implementation of the Gaussian bilateral filter is rather time consuming and is thus too slow for real-time applications. Consequently, many research papers have been devoted to accelerating the computation of the bilateral filter, and many publications can be found during the past decade.

2.1. Kernel Separation. Pham and Vliet proposed a method to approximate the 2-D data-adaptive convolution using two 1-D data-adaptive convolutions [18]. The computational complexity is reduced from $O(\sigma_r^2)$ operations per pixel to $O(\sigma_r)$ operations per pixel, where σ_r denotes the radius of the effective spatial support. This method performs well in regions with simple structure. However, it does not produce satisfactory results in textured regions.

2.2. Piecewise Linear Approximation. Durand and Dorsey proposed a piecewise linear approximation of the bilateral filter in [9]. The authors first quantize the original intensity into several segments. Then for each segment i_k , a range-weighted image pair (J_k, W_k) is calculated. Afterwards, a linearly filtered result u_k for each i_k is generated by linear filtering with the corresponding range-weighted image pair, which can be formulated as follows:

$$u_k(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in N(\mathbf{x})} K_s(\mathbf{x} - \mathbf{y}) J_k(\mathbf{y})}{\sum_{\mathbf{y} \in N(\mathbf{x})} K_s(\mathbf{x} - \mathbf{y}) W_k(\mathbf{y})}, \quad (2)$$

where $W_k(\mathbf{y}) = K_r(i_k - v(\mathbf{y}))$ and $J_k(\mathbf{y}) = W^k(\mathbf{y})v(\mathbf{y})$. Then the final result is obtained by linearly interpolating between two closest filtered images; that is, for $v(\mathbf{x}) \in [i_k, i_{k+1}]$, the filtered output at \mathbf{x} is given as follows:

$$u^{\text{BF}}(\mathbf{x}) = \frac{(i_{k+1} - v(\mathbf{x})) u_k(\mathbf{x}) + (v(\mathbf{x}) - i_k) u_{k+1}(\mathbf{x})}{i_{k+1} - i_k}. \quad (3)$$

The computational complexity is dramatically reduced since the implementation requires only $O(\log\sigma_r)$ operations

per pixel using fast Fourier transform (FFT) to compute each u_k . It is further accelerated by performing a subsampling in the spatial domain.

Recently, based on Durand's formulation, Yang et al. [23] employed Deriche's recursive approximation of Gaussian kernel [28] to calculate the spatial filtering, which can be implemented in constant time and can be realized in real time in some graphic processing units (GPU).

2.3. 3-D Bilateral Grid Manipulations. The piecewise linear bilateral filter is later generalized and further accelerated by Paris et al. in [19]. The authors express the filtering in a higher-dimensional space where another dimension for the intensity is added to the original spatial dimensions. Then the bilateral filtering can be expressed as simple linear convolutions with a vector-valued image in this augmented space, followed by a point-by-point division. With the new representation of the data, simple criteria are derived for subsampling the data to achieve significant acceleration of the bilateral filter. With the same computational time, this method is more accurate than the piecewise linear bilateral filter by Durand and Dorsey in [9].

Built upon the technique by Paris et al. in [19], Chen et al. [11] extend the higher-dimensional approach and introduce a new compact data structure—the bilateral grid. This enables a variety of fast edge-preserving image processing applications. The authors parallelize the bilateral-grid algorithm in modern graphics hardware with graphic processing units (GPU). The GPU-based implementation is two orders of magnitude faster than the equivalent CPU-based implementations. This enables the real-time edge-preserving manipulations on high-definition images. The authors demonstrate the use of the bilateral-grid method on a variety of applications including image editing, high-dynamic-range tone mapping, and image enhancement.

The computational complexity of the bilateral-grid methods is $O(1 + |\mathcal{R}|/\sigma_s\sigma_r)$ operations per pixel, where \mathcal{R} is the dynamic range of the intensity and $|\mathcal{R}|$ is the number of grids occupied by the range dimension. This results in a paradoxical property that the algorithm proceeds faster when the size of the spatial kernel is larger, due to larger subsampling rate. However, the exact output is dependent on the phase of subsampling. Furthermore, the operations with small σ_s and σ_r require fine subsampling. This requires larger memory and more computation time. Furthermore, the aggregation of the final result relies on a trilinear interpolation among the results on the grids. Better accuracy can be obtained using higher-order interpolations. Consequently, there is a tradeoff between the quality and the computational cost using this type of bilateral filters.

2.4. Histogram-Based Approximation Using Box Spatial Kernel. Weiss developed an iterative method to compute local histogram with $O(\log \sigma_r)$ complexity in [20]. He later demonstrated that such fast local histogram computation can be applied for the $O(\log \sigma_r)$ bilateral filter when the spatial kernel is a uniform box. Similar to Weiss' work, Porikli proposed an $O(1)$ bilateral filter [21] based on his earlier

work on $O(1)$ integral histogram [22]. The bilateral filter with single-box spatial kernel is formulated as follows:

$$u^{\text{bBF}}(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in B(\mathbf{x})} K_r(v(\mathbf{x}) - v(\mathbf{y})) v(\mathbf{y})}{\sum_{\mathbf{y} \in B(\mathbf{x})} K_r(v(\mathbf{x}) - v(\mathbf{y}))}, \quad (4)$$

where $B(\mathbf{x})$ denotes the box with uniform weight centered at \mathbf{x} . Then the above formulation can be further expressed in terms of local histograms as follows:

$$u^{\text{bBF}}(\mathbf{x}) = \frac{\sum_{p=1}^P H_{\mathbf{x}}(I_p) K_r(v(\mathbf{x}) - I_p) I_p}{\sum_{p=1}^P H_{\mathbf{x}}(I_p) K_r(v(\mathbf{x}) - I_p)}, \quad (5)$$

where P is the number of histogram bin, I_p denotes the intensity level of the p th histogram bin, and $H_{\mathbf{x}}(I_p)$ indicates the number of pixels in the p th histogram bin within the spatial neighborhood centered at \mathbf{x} . Then the computational time is constant with $O(1)$ integral histogram technique [22]. And the calculation can be further accelerated by the quantizing the intensity into a small number of histogram bins.

Based on Porikli's single-box bilateral filter, Gunturk proposed a method to approximate the bilateral filter with arbitrary spatial and range kernels using a weighted sum of multiple single-box bilateral filters [26], which can be formulated as follows:

$$u^{\text{BF}}(\mathbf{x}) \approx \sum_{m=1}^M \hat{k}_m u_m^{\text{bBF}}(\mathbf{x}) \doteq u^{\text{mbBF}}(\mathbf{x}). \quad (6)$$

The coefficients $\{\hat{k}_m\}$ are then obtained by solving a set of linear equations which minimize the squared error between the discrete forms of

$$K_s(\mathbf{x} - \mathbf{y}) K_r(v(\mathbf{x}) - v(\mathbf{y})), \quad (7)$$

$$\sum_{m=1}^M k_m B_m(\mathbf{x} - \mathbf{y}) K_r(v(\mathbf{x}) - v(\mathbf{y})).$$

This improvement provides better approximation of the bilateral filter with arbitrary spatial and range kernels by costing a little computation time compared with the single-box bilateral filter. However, further study shows that the approximation given by (6) only performs well when the size of spatial kernel is relatively small. As the radius of the spatial support increases, more single-box bilateral filters should be combined to guarantee the approximation accuracy according to Gunturk's scheme. Thus the computational complexity will be unbearable for efficient applications if the size of the spatial kernel is too large.

2.5. Accelerations Based on Series Expansion of the Range Kernel. In [21], Porikli proposes another $O(1)$ bilateral filter with arbitrary spatial kernel and Gaussian range kernel, which employs Taylor series expansion of the range kernel. Then the bilateral filtering is decomposed into a set of spatial filtering steps with the image series computed prior to the convolution. To guarantee a constant-time processing,

the author proposes to subsample the separable 1D linear spatial filters to a constant number of taps asymmetrically. However, the accuracy is poor using a low order Taylor expansion when the variance of Gaussian range kernel is small. Moreover, the range weight blows up when the intensity difference is too large with respect to the variance. Hence, the decaying monotonicity for an admissible kernel is broken.

As an extension of Porikli's polynomial bilateral filter in [21], Chaudhury et al. propose a constant-time bilateral filter with trigonometric range kernels in [24]. Based on the fact that the raised cosine series converges to a Gaussian, the trigonometric range kernel is applied to approximate Gaussian range kernel. The authors show that the bilateral filter using trigonometric series is more accurate than the one using Taylor series with the same number of terms. The computation time of the trigonometric bilateral filter is constant with respect to the variance of the spatial kernel. However, the complexity is $O(|\mathcal{R}|^2/\sigma_r^2)$ with respect to the variance of the range kernel, where $|\mathcal{R}|$ denotes the number of quantization levels of the intensity and σ_r is the variance of the range kernel. Consequently, the bilateral filtering with small σ_r will be time consuming using trigonometric range kernel.

More recently in [25], the trigonometric bilateral filter is accelerated using truncations when σ_r is small and further extended to a larger class of data-dependent filters including the well-known nonlocal means filter.

3. Fast Implementation of the Bilateral Filter Using Sparse Approximation with Fixed Number of Boxes

In this paper, we propose a method to approximate arbitrary spatial kernel with multiple boxes, which can then be leveraged for the constant-time implementation of the bilateral filter with arbitrary spatial and range kernels. Let L be the radius of the spatial support of the given spatial kernel K_s applied in the bilateral filter. Then all the candidate boxes together form a series $\{B_l\}$, where l is the radius of the box B_l and $l = 0, 1, 2, \dots, L$. For arbitrary K_s , it can be approximated using the weighted sum of all the candidate boxes, which is formulated as follows:

$$K_s(\mathbf{x} - \mathbf{y}) \approx \sum_{l=0}^L k_l B_l(\mathbf{x} - \mathbf{y}). \quad (8)$$

Because of the symmetry and monotonicity of any admissible spatial kernel, it is possible to find a real positive series $\{k_l\}$ that minimizes the following squared error:

$$\left\| K_s - \sum_{l=0}^L k_l B_l \right\|_2^2, \quad (9)$$

where the spatial dependency is omitted for simplicity.

3.1. Approximating Arbitrary Spatial Kernel with Fixed Number of Boxes. For that the number of boxes is great when

L is large, the computational cost will become unbearable for real-time applications. In order to handle this problem, we add a constraint that the number of boxes used in the approximation should not be larger than a preset number N , such that the constrained minimization of the squared error can be further formulated as follows:

$$\min_{k_l} \left\| K_s - \sum_{l=0}^L k_l B_l \right\|_2^2, \quad \text{s.t. } \#\{l \mid k_l \neq 0\} \leq N. \quad (10)$$

The left part of the above formulation is again the least squared error norm. And the right part is employed to limit the number of boxes used in the approximation, which denotes the number of boxes that have nonzero coefficients. For any $l \in [0, L]$, we align the center of the corresponding box with that of K_s and pad it with zeros up to the same size as K_s . Then the columns of each padded box are concatenated to form a column vector \mathbf{b}_l . We then put these column vectors together to form a matrix \mathbf{B} of size $S \times (L + 1)$, where $S = (2L + 1)^2$ is the number of elements in K_s . Correspondingly, we concatenate the columns of K_s to form a column vector \mathbf{q} . By defining \mathbf{k} as a column vector containing all the coefficients k_l , the optimization problem given in (10) can be reformulated as follows:

$$\hat{\mathbf{k}} = \arg \min_{\mathbf{k}} \|\mathbf{q} - \mathbf{B}\mathbf{k}\|^2, \quad \text{s.t. } \|\mathbf{k}\|_0 \leq N, \quad (11)$$

where $\|\mathbf{k}\|_0$ is the L_0 norm of the vector \mathbf{k} , which denotes the number of non-zero elements in \mathbf{k} .

One should be very familiar with the formulation given by (11), since it is a well-known sparse representation problem. This formulation searches the sparse representation of the signal \mathbf{q} using the known dictionary \mathbf{B} . This sparse approximation problem, which is known to be NP-hard, can be efficiently solved using several available techniques, including Orthogonal Matching Pursuit (OMP) [27, 29, 30], Basis Pursuit (BP) [31, 32], and FOCUSS [33].

In this paper, we employ the efficient OMP algorithm by Rubinstein et al. in [27], namely, Batch-OMP, to solve (11). After we find out the radiuses and coefficients of the boxes corresponding to the given spatial kernel and the given number of boxes, the further computation of the bilateral filter can be given as follows:

$$\begin{aligned} u^{\text{BF}}(\mathbf{x}) &\approx u^{\text{obBF}}(\mathbf{x}) \\ &= \frac{\sum_{\mathbf{y}} \left(\sum_{l|k_l \neq 0} k_l B_l(\mathbf{x} - \mathbf{y}) \right) K_r(v(\mathbf{x}) - v(\mathbf{y})) v(\mathbf{y})}{\sum_{\mathbf{y}} \left(\sum_{l|k_l \neq 0} k_l B_l(\mathbf{x} - \mathbf{y}) \right) K_r(v(\mathbf{x}) - v(\mathbf{y}))}. \end{aligned} \quad (12)$$

With this respect, we can benefit from the integral map technique to efficiently calculate the spatial convolution in constant time. It is worth mentioning that N in (10) and (11) is an input parameter defined by the users. For a given σ_s , the more the number of boxes used is, the better the approximation accuracy can be achieved. But more computational time is needed for more boxes. Thus there is a tradeoff between the approximation accuracy and the computational efficiency

with our method. Generally, we can find a minimum N that guarantees the acceptable quality of the approximation, which will be discussed in detail in the next section.

3.2. Application to the Histogram-Based Bilateral Filters. Now having that an arbitrary spatial kernel is approximated with a preset number of boxes, we first exploit its use in the histogram-based bilateral filters, which is originally developed by Weiss in [20], later accelerated by Porikli in [21], and recently extended by Gunturk in [26].

Firstly, let us denote the solution of (11) using the notation $\{k_n\}$, which contains N non-zero coefficients for N boxes with different radiuses selected by the Batch-OMP algorithm. The subscripts n 's now are not the radiuses but the indices of the selected boxes; that is, $n = 1, 2, \dots, N$. Then the proposed bilateral filter with multiple-box spatial kernel given in (12) can be reformulated as follows:

$$\begin{aligned} u^{\text{obBF}}(\mathbf{x}) &= \frac{\sum_{\mathbf{y}} \left(\sum_{n=1}^N k_n B_n(\mathbf{x} - \mathbf{y}) \right) K_r(v(\mathbf{x}) - v(\mathbf{y})) v(\mathbf{y})}{\sum_{\mathbf{y}} \left(\sum_{n=1}^N k_n B_n(\mathbf{x} - \mathbf{y}) \right) K_r(v(\mathbf{x}) - v(\mathbf{y}))} \\ &= \frac{\sum_{n=1}^N k_n \sum_{\mathbf{y}} B_n(\mathbf{x} - \mathbf{y}) K_r(v(\mathbf{x}) - v(\mathbf{y})) v(\mathbf{y})}{\sum_{n=1}^N k_n \sum_{\mathbf{y}} B_n(\mathbf{x} - \mathbf{y}) K_r(v(\mathbf{x}) - v(\mathbf{y}))}. \end{aligned} \quad (13)$$

Following the single-box histogram-based bilateral filter by Weiss [20] and Porikli [21], (13) can be further formulated in terms of local histogram as follows:

$$u^{\text{obBF}}(\mathbf{x}) = \frac{\sum_{n=1}^N k_n \sum_{p=1}^P I_p H_{\mathbf{x},n}(I_p) K_r(v(\mathbf{x}) - I_p)}{\sum_{n=1}^N k_n \sum_{p=1}^P H_{\mathbf{x},n}(I_p) K_r(v(\mathbf{x}) - I_p)}. \quad (14)$$

Prior to the filtering process, the intensity is quantized into P bins, resulting in an intensity series $\{I_p\}_1^P$. Then the corresponding P frames of range weight $\{K_r(v(\mathbf{x}) - I_p)\}$ are calculated. Based on the work in [22], an integral histogram is established. Then N local histogram maps are calculated in constant time using the same integral histogram. Compared with the single-box bilateral filter, the proposed method requires $N - 1$ additional local histogram calculations, which can be efficiently obtained using the integral histogram technique. The calculations of N times of box filtering for both the numerator and dominator in (14) can be parallelized for more efficient implementation.

3.3. Application to the Interpolation-Based Bilateral Filters. Since Durand's piecewise linear bilateral filter [9], Yang's extension [23], and also Paris' method [19] employ an interpolation in pooling the filtering results, we call them together the interpolation-based bilateral filters for convenience. As the origin of this type of bilateral filters, Durand's method is discussed in this paper for demonstrating the application of the proposed approximation to this type of bilateral filters. Similar adaptation can be easily made for the application to Yang's method.

Let us replace the spatial kernel K_s in (2) by Durand and Dorsey using the weighted sum of multiple boxes, and the calculation of u_k is formulated as follows:

$$\begin{aligned} u_k(\mathbf{x}) &\approx \frac{\sum_{\mathbf{y}} \left(\sum_{n=1}^N k_n B_n(\mathbf{x} - \mathbf{y}) \right) J_k(\mathbf{y})}{\sum_{\mathbf{y}} \left(\sum_{n=1}^N k_n B_n(\mathbf{x} - \mathbf{y}) \right) W_k(\mathbf{y})} \\ &= \frac{\sum_{n=1}^N k_n \sum_{\mathbf{y}} B_n(\mathbf{x} - \mathbf{y}) J_k(\mathbf{y})}{\sum_{n=1}^N k_n \sum_{\mathbf{y}} B_n(\mathbf{x} - \mathbf{y}) W_k(\mathbf{y})}. \end{aligned} \quad (15)$$

Given an image, it is first divided into a preset number of intensity segments, forming an image series $\{i_k\}$. Correspondingly, a range-weighted image series $\{J_k(\mathbf{y})\}$ and a weight series $\{W_k(\mathbf{y})\}$ are computed. Afterwards, the corresponding integral maps of these two image series are established. The filtered result u_k for each image segment i_k is computed according to (15) with $O(1)$ complexity. The final result is obtained using an interpolation, as given in (3) for an example. The running time of this bilateral filter is constant with respect to the size of the spatial kernel.

3.4. Applications to the Series-Based Bilateral Filters. Since a convolution with spatial kernels is also involved in the series-based bilateral filters, the proposed sparse approximation is also possible to be applied in this type of accelerations. The detailed adaptation for the series-based bilateral filters is omitted in this paper, which can be easily derived by the readers.

4. Experiments and Comparisons

The proposed algorithm is implemented in MATLAB 2010b under Windows XP SP3 (32 bits), in a PC computer with Pentium Dual-Core T4200 @ 2.00 GHz each and 3 GB RAM. The proposed histogram-based bilateral filter with multiple-box spatial kernel is tested and discussed in this section. The experiments on approximating the bilateral filter with Gaussian spatial and range kernels provide the readers with a glance at the validity and the effectiveness of the proposed method. The application to other types of the acceleration of the bilateral filter can be easily implemented and evaluated under the guidance provided in the previous section. But it is not given here for saving the length of this paper. In the following text of this section, we evaluate the performance of the proposed histogram-based bilateral filter in both qualitative and quantitative aspects.

Before the evaluation, we briefly introduce the basic settings of the experiments discussed in the following text. Four standard test images are employed, which are 8 bit grayscale images of size 512×512 , as shown in Figure 1. The numerical results given in the following texts are all produced using the standard test images, unless otherwise noticed.

4.1. Qualitative Demonstration. For the qualitative demonstration, the proposed histogram-based method, together with Porikli's single-box bilateral filter and Gunturk's bilateral filter, is employed to approximate the standard bilateral filter

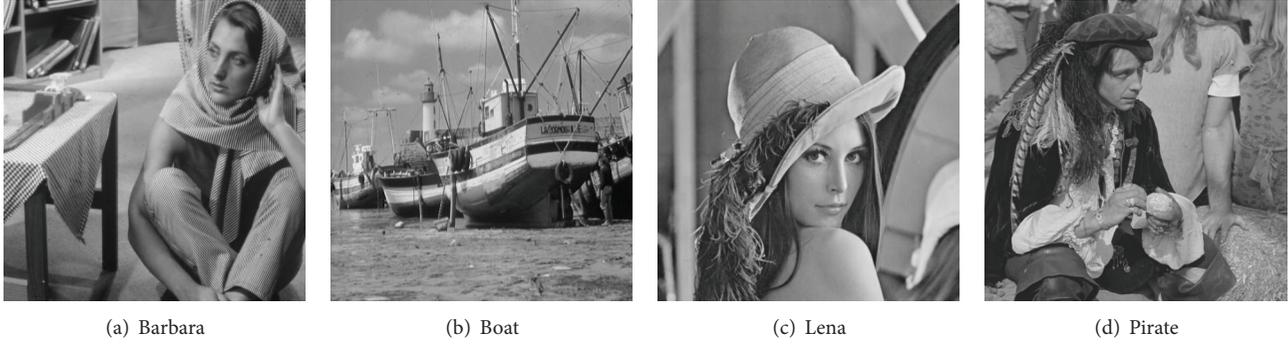


FIGURE 1: Four standard test images used in the experiments in this paper. They are 8-bit grayscale images of size 512×512 .

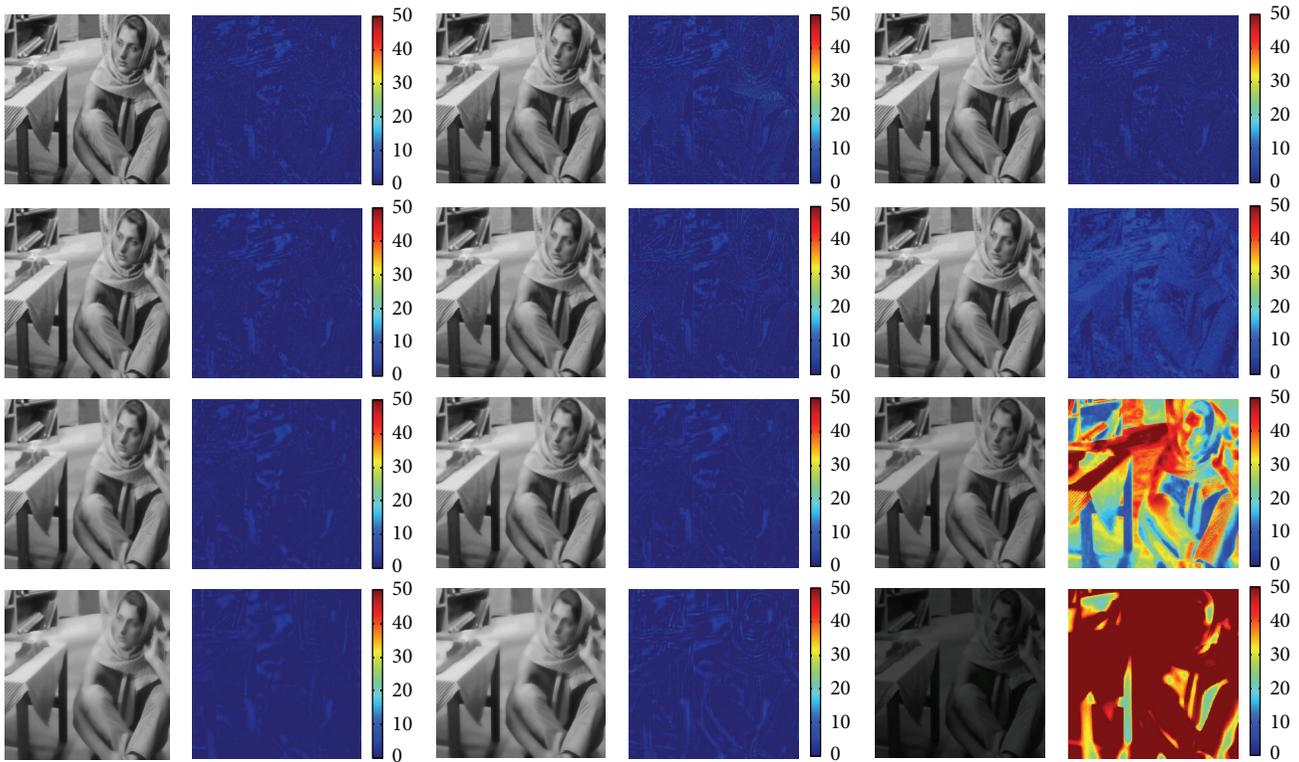


FIGURE 2: Barbara: a demonstration of the visual quality of the approximation results. First column: results obtained using the proposed method, where $N = 5$. Second column: the color-coded absolute error maps corresponding to the images in the first column. Third column: results obtained using Porikli's single-box bilateral filter with optimal spatial parameter. Fourth column: the color-coded absolute error maps corresponding to the images in the third column. Fifth column: results obtained using Gunturk's bilateral filter, where $M = 5$. Last column: the color-coded absolute error maps corresponding to the images in the fifth column. First row: images are obtained with $\sigma_s = 1.2$. Second row: images are obtained with $\sigma_s = 1.8$. Third row: images are obtained with $\sigma_s = 3.0$. Fourth row: images are obtained with $\sigma_s = 6.0$. For all these results, $\sigma_r = 50$ and $P = 16$. The ranges of the color-coded maps are uniformly limited to $[0, 50]$ to facilitate visual comparison.

with Gaussian spatial and range kernels. The results are illustrated in Figure 2. As shown in Figure 2, the images in the first, third, and fifth columns are, respectively, the results of the proposed histogram-based bilateral filter, the results of Porikli's single-box bilateral filter [21], and the results of Gunturk's bilateral filter [26]. The color-coded maps in the second, fourth, and sixth columns are the corresponding absolute error between the images, respectively, in the first, third, and fifth columns and the results of the standard

Gaussian bilateral filter, respectively, with the same input parameters. The variances of the spatial kernels are $\sigma_s = 1.2, 1.8, 3.0,$ and $6.0,$ respectively, for the four rows. The variances of the range kernels are the same; that is, $\sigma_r = 50$. The number of boxes for the proposed method is $N = 5$ in producing the images in the first column. Correspondingly, the number of box bilateral filters used in Gunturk's method is $M = 5$ in producing the results in the fifth column. For Porikli's method, the results are obtained with optimal spatial

TABLE 1: Comparison of the PSNR values.

σ_s	Barbara			Boat			Lena			Pirate		
	Ours	[21]	[26]									
0.90	41.90	40.34	41.62	43.15	41.89	42.92	41.45	40.89	41.43	42.68	41.70	42.60
1.20	43.28	39.21	43.10	44.70	40.10	44.37	42.58	41.06	42.53	44.15	40.83	44.04
1.50	44.27	41.68	43.50	45.70	43.26	43.92	43.35	42.61	42.85	45.17	43.52	44.36
1.80	45.05	41.98	37.89	46.44	40.95	37.08	43.90	42.01	37.61	45.90	41.98	38.48
2.40	46.14	43.89	24.81	47.35	44.05	24.11	44.75	43.37	24.55	46.91	44.27	25.35
3.00	46.84	44.79	18.25	47.85	44.29	17.60	45.51	44.10	17.97	47.69	44.94	18.78
3.60	47.51	45.28	14.56	48.41	44.39	13.93	46.09	44.57	14.27	48.36	45.25	15.09
4.50	48.07	45.00	11.52	48.85	44.43	10.90	46.75	44.53	11.23	48.93	44.93	12.05
6.00	48.88	44.74	9.16	49.33	44.40	8.53	47.58	44.52	8.86	49.66	44.56	9.69
7.50	49.34	44.25	8.06	49.62	44.11	7.42	48.19	44.47	7.75	50.21	44.25	8.59
9.00	49.67	44.89	7.46	49.92	44.27	6.83	48.61	45.22	7.16	50.51	45.23	7.99

TABLE 2: Comparison of the FSIM values.

σ_s	Barbara			Boat			Lena			Pirate		
	Ours	[21]	[26]	Ours	[21]	[26]	Ours	[21]	[26]	Ours	[21]	[26]
0.90	0.9874	0.9859	0.9873	0.9928	0.9921	0.9927	0.9830	0.9810	0.9829	0.9911	0.9902	0.9911
1.20	0.9902	0.9893	0.9902	0.9943	0.9911	0.9941	0.9862	0.9864	0.9862	0.9931	0.9908	0.9930
1.50	0.9924	0.9914	0.9924	0.9954	0.9949	0.9952	0.9888	0.9879	0.9889	0.9946	0.9940	0.9946
1.80	0.9939	0.9922	0.9939	0.9962	0.9901	0.9959	0.9907	0.9896	0.9909	0.9956	0.9913	0.9956
2.40	0.9960	0.9942	0.9929	0.9972	0.9949	0.9941	0.9934	0.9911	0.9910	0.9969	0.9948	0.9944
3.00	0.9973	0.9953	0.9817	0.9978	0.9941	0.9834	0.9955	0.9931	0.9820	0.9977	0.9950	0.9849
3.60	0.9979	0.9956	0.9589	0.9981	0.9931	0.9629	0.9964	0.9939	0.9629	0.9982	0.9947	0.9658
4.50	0.9984	0.9954	0.9137	0.9984	0.9927	0.9235	0.9973	0.9939	0.9248	0.9985	0.9944	0.9280
6.00	0.9988	0.9956	0.8427	0.9985	0.9931	0.8631	0.9979	0.9940	0.8660	0.9988	0.9942	0.8697
7.50	0.9991	0.9956	0.7928	0.9987	0.9932	0.8198	0.9984	0.9943	0.8246	0.9990	0.9944	0.8287
9.00	0.9992	0.9962	0.7602	0.9989	0.9935	0.7916	0.9987	0.9958	0.7979	0.9990	0.9952	0.8021

parameter derived by Gunturk in [26]; that is, $r = [1.4\sigma_s]$. For producing all the results shown in Figure 2, $P = 16$. The ranges of the absolute error maps shown here are uniformly limited to $[0, 50]$ for better illustrating the difference of the performance of the three methods.

It is shown in Figure 2 that the proposed histogram-based bilateral filter produces less error than the other two methods when varying σ_s . Hence, better accuracy is achieved in approximating the standard bilateral filter. Note that the error maps in the fourth column show that Porikli's method with optimal box performs better than Gunturk's method when σ_s is sufficiently large. Especially noted in the fifth column, Gunturk's method performs well with small σ_s , but the result images gradually become darker as σ_s increases with fixed M . The performance of Gunturk's bilateral filter can be improved by increasing M . However, the computational cost will increase as well. The numerical results on this experiment are also included in Tables 1 and 2.

4.2. Quantitative Evaluation. We quantitatively evaluate the proposed method in three aspects, including the running time on our system with nonoptimized MATLAB codes, the quality in approximating the standard bilateral filter according to some previously established criteria, and the

minimum number of boxes to guarantee sufficient accuracy compared with the minimum number of box bilateral filters in Gunturk's method.

4.2.1. Computational Efficiency. We test our method by varying values of N and P and record the running time. Experiments show that the proposed method costs about 2.36 s for a 512×512 grayscale image when $N = 5$ and $P = 16$. And every additional histogram bin costs about 0.04 s when $N = 5$, and every additional box costs about 0.36 s when $P = 16$. The local histogram calculation costs 0.34 s each time on our system, which is the most time-consuming part of the nonoptimized codes for every additional box. Further acceleration of the local histogram calculation should be done for more efficient implementation of the proposed method.

Experiments show that the running time of the proposed method is about 0.1 s more than that of Gunturk's method with $N = M$ and $P = 16$. It is because the sparse approximation algorithm in the proposed method costs about 0.1 s more than the linear problem solver in Gunturk's algorithm. However, the computation of any additional box costs as much time as an additional single-box bilateral filter. As a result, the running time of the rest codes of the two algorithms is almost the same.

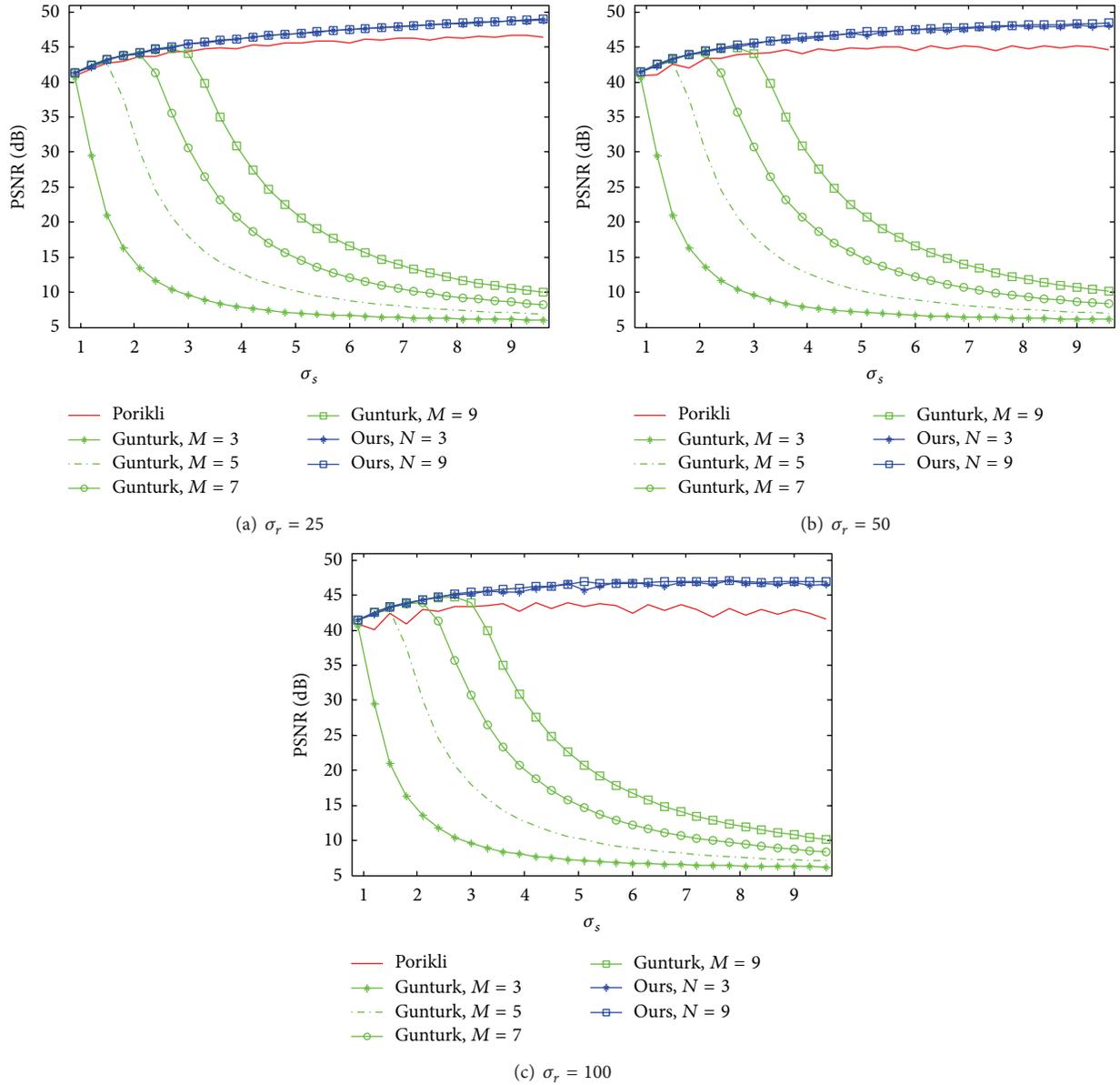


FIGURE 3: The PSNR values of the results with respect to σ_s . These results are calculated on “Lena” in approximating the Gaussian bilateral filter, where $P = 16$.

4.2.2. Approximation Accuracy. The two previously established criteria are employed in quantitatively evaluating the approximation quality in this paper. Following the previous work, the PSNR is employed. Recently, Zhang et al. propose a feature similarity index (FSIM) in [34] for image quality assessment with reference. It is one of the state-of-the-art methods for predicting the similarity between the test and reference images. In order to provide a more comprehensive evaluation, the FSIM is also applied for evaluating the quality of the results.

Given the result images of the standard bilateral filter with different parameter settings, the PSNR and FSIM values of the results by the methods to be evaluated are calculated.

The PSNR and FSIM values on four standard test images of the proposed histogram-based method together with that of Porikli's and Gunturk's methods are, respectively, listed in Tables 1 and 2, where $\sigma_r = 50$, $P = 16$, and $N = M = 5$. The best scores among the three methods for each individual σ_s and each test image are shown in boldface. We can see from these two tables that our method outperforms the other two according to both criteria. And it is more robust with respect to the change of σ_s with a fixed N . Besides, Gunturk's bilateral filter performs well when σ_s is small. However, both PSNR and FSIM values of Gunturk's method decrease quickly as σ_s increases when M is fixed. It is because the result images become darker as σ_s increases for a fixed M , as can be

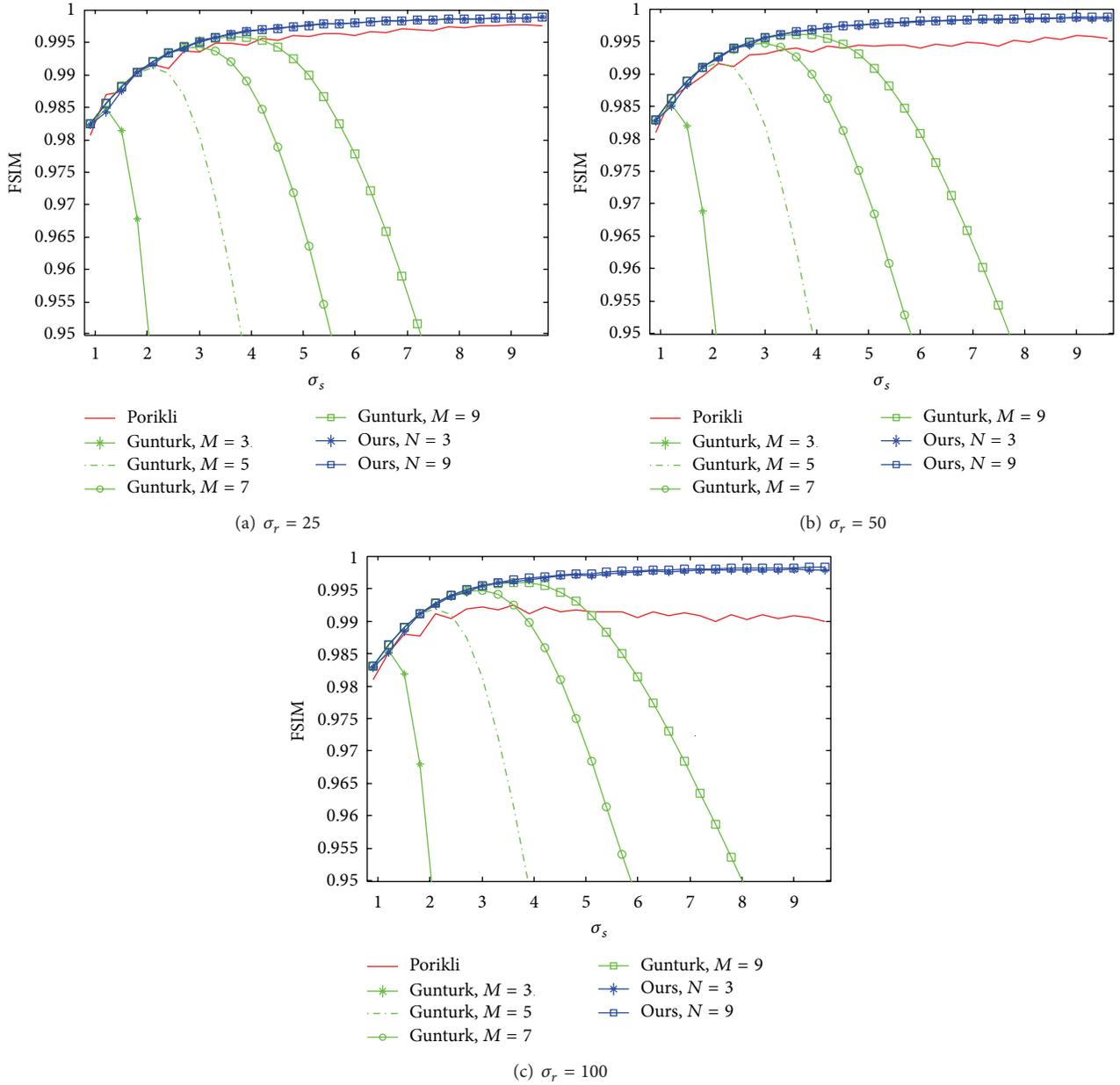


FIGURE 4: The FSIM values of the results with respect to σ_s . These results are calculated on “Lena” in approximating the Gaussian bilateral filter, where $P = 16$.

illustrated in Figure 2. Larger M is required for better quality in approximating Gaussian bilateral filter with larger σ_s using Gunturk’s method.

In order to provide a more intuitive illustration, we plot the PSNR and FSIM values on “Lena,” respectively, in Figures 3 and 4 with respect to σ_s . As can be seen in these two figures, the proposed histogram-based bilateral filter has higher PSNR and FSIM values compared with the other two methods over a large range of σ_s . Besides, the value of N does not affect the performance too much. As a result, only a small number of boxes are sufficient for a vast range of σ_s using the proposed method. On the contrary, for a fixed M , the scores given by both principles

of Gunturk’s method decay quickly as σ_s increases. Using larger M is helpful in producing satisfactory results; however, as discussed before, the computational efficiency will be degraded as a consequence. Although small σ_s is sufficient for most denoising applications, large σ_s is usually required in many other applications including high-dynamic-range tone mapping [9], image enhancement [14], stereo vision [17], and dehazing [35]. Consequently, the computational cost will be unbearable for efficient implementation combining a large number of single-box bilateral filters.

Furthermore, as can be seen in Figures 3 and 4, our histogram-based bilateral filter is apparently more robust than the other two methods with respect to σ_r . On the

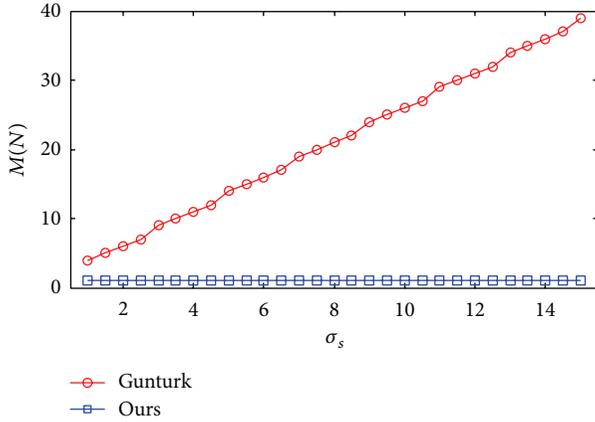


FIGURE 5: Minimum number of boxes required to guarantee PSNR > 40 dB, where $P = 16$ and $\sigma_r = 50$.

contrary, as the value of σ_r increases, the performance of Porikli's single-box bilateral filter becomes less stable with respect to the change of σ_s .

4.2.3. Minimum Number of Box. For Gunturk's method, larger M leads to better approximation accuracy. Thus sufficient single-box bilateral filters should be combined to guarantee an acceptable accuracy. Correspondingly, as mentioned in the previous section, sufficient boxes should also be used in the proposed histogram-based bilateral filter. For the acceptable quality threshold, it is often considered as visually undistinguishable when $\text{PSNR} \geq 40$ dB [21]. We experiment on a standard test image, namely, "Lena," shown as Figure 1(c). In order to find out the minimum N for the proposed histogram-based bilateral filter and the corresponding minimum M for Gunturk's method with respect to the scale of the spatial kernel, we vary the value of σ_s and keep other parameters unchanged. For each given σ_s , we gradually increase both numbers and calculate the PSNR values of the results, with respect to the results of the standard Gaussian bilateral filter. Then the minimum numbers which enable the quality to exceed the acceptable threshold for both algorithms can be obtained. The minimum N and the minimum M with respect to σ_s are shown in Figure 5. It is clearly seen that the minimum M for Gunturk's method increases as σ_s increases. On the contrary, the minimum N for the proposed method is much smaller than the minimum M for Gunturk's method. And it is nearly constant with respect to σ_s . $P = 16$ and $\sigma_r = 50$ for both methods in this experiment. It is shown that the proposed histogram-based bilateral filter is more efficient to obtain sufficient accuracy, and the performance is more robust to the change of the spatial scale of the filter.

Finally, it is worth noting that, when $N = 1$, the proposed histogram-based bilateral filter is exactly the same as Porikli's single-box bilateral filter with optimal spatial parameter.

5. Conclusions and Discussions

We have presented a method for approximating arbitrary spatial kernel using a fixed number of boxes based on sparse approximation techniques. With applications to the acceleration of the bilateral filter, the proposed method can be leveraged for a broad class of constant-time bilateral filters with arbitrary spatial and range kernels. Once the parameter of the spatial kernel and the number of boxes are given, the radiuses and the coefficients of the boxes are determined by the Batch-OMP algorithm. Hence they can be computed ahead of the filtering process. The application to the histogram-based $O(1)$ bilateral filter is demonstrated in this paper, followed by a number of convictive experiments. Results tell that the proposed histogram-based bilateral filter has better accuracy compared with other histogram-based bilateral filters in approximating the standard bilateral filter. Meanwhile, the performance of the proposed histogram-based bilateral filter is robust with respect to the parameters of the filter kernels. And a small number of boxes are sufficient to achieve satisfactory accuracy for a large range of the spatial parameters. With a very little more computational cost, the accuracy of the proposed histogram-based bilateral filter is better than the previous histogram-based ones.

The proposed $O(1)$ bilateral filter with arbitrary spatial and range kernels can be parallelized for real-time implementation in GPUs. Moreover, it is possible for the proposed method to be extended to the bilateral grid. 3-D cubic boxes can be used to approximate 3-D arbitrary spherical filter kernels in the augmented data space. With the proposed method, higher-dimensional manipulations are also possible in this direction.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Dr. Jian Li for his insightful discussion over the mathematical deductions and the implementation of the algorithms. This work is supported by the National Natural Science Foundation of China, under Grant 90820302.

References

- [1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the 6th IEEE International Conference on Computer Vision*, pp. 839–846, January 1998.
- [2] S. M. Smith and J. M. Brady, "SUSAN—a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.
- [3] L. P. Yaroslavsky, *Digital Picture Processing—An Introduction*, Springer, Berlin, Germany, 1985.
- [4] J.-S. Lee, "Digital image smoothing and the sigma filter," *Computer Vision, Graphics and Image Processing*, vol. 24, no. 2, pp. 255–269, 1983.

- [5] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1151, 2002.
- [6] E. P. Bennett, J. L. Mason, and L. McMillan, "Multispectral bilateral video fusion," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1185–1194, 2007.
- [7] M. Zhang and B. K. Gunturk, "Multiresolution bilateral filtering for image denoising," *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2324–2333, 2008.
- [8] A. Wong, "Adaptive bilateral filtering of image signals using local phase characteristics," *Signal Processing*, vol. 88, no. 6, pp. 1615–1619, 2008.
- [9] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '02)*, pp. 257–266, San Antonio, Tex, USA, July 2002.
- [10] S. Bae, S. Paris, and F. Durand, "Two-scale tone management for photographic look," in *Proceedings of the ACM SIGGRAPH (SIGGRAPH '06)*, pp. 637–645, Boston, Mass, USA, August 2006.
- [11] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Transactions on Graphics*, vol. 26, no. 3, Article ID 1276506, pp. 1–9, 2007.
- [12] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," in *Proceedings of the ACM SIGGRAPH*, pp. 673–678, Los Angeles, Calif, USA, August 2004.
- [13] M. Elad, "Retinex by two bilateral filters," in *Proceedings of Scale-Space and PDE Methods in Computer Vision*, vol. 3459, pp. 217–229, Hofgeismar, Germany, April 2005.
- [14] B. Zhang and J. P. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 664–678, 2008.
- [15] R. Ramanath and W. E. Snyder, "Adaptive demosaicking," *Journal of Electronic Imaging*, vol. 12, no. 4, pp. 633–642, 2003.
- [16] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi, "Bilateral filtering-based optical flow estimation with occlusion detection," in *Proceedings of the European Conference on Computer Vision (ECCV '06)*, pp. 211–224, May 2006.
- [17] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 492–504, 2009.
- [18] T. Q. Pham and L. J. van Vliet, "Separable bilateral filtering for fast video preprocessing," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '05)*, pp. 1–4, Amsterdam, Netherlands, July 2005.
- [19] S. Paris, F. Dur, and A. fast, "A fast approximation of the bilateral filter using a signal processing approach," in *Proceedings of the European Conference on Computer Vision (ECCV '06)*, pp. 829–836, Graz, Austria, May 2006.
- [20] B. Weiss, "Fast median and bilateral filtering," in *Proceedings of the ACM SIGGRAPH*, pp. 519–526, Boston, Mass, USA, August 2006.
- [21] F. Porikli, "Constant time $O(1)$ bilateral filtering," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [22] F. Porikli, "Integral histogram: a fast way to extract histograms in Cartesian spaces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 829–836, San Diego, Calif, USA, June 2005.
- [23] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time $O(1)$ bilateral filtering," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 557–564, June 2009.
- [24] K. N. Chaudhury, D. Sage, and M. Unser, "Fast $O(1)$ bilateral filtering using trigonometric range kernels," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3376–3382, 2011.
- [25] K. N. Chaudhury, "Acceleration of the shiftable $O(1)$ algorithm for bilateral filtering and nonlocal means," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1291–1300, 2013.
- [26] B. K. Gunturk, "Fast bilateral filter with arbitrary range and domain kernels," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2690–2696, 2011.
- [27] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Tech. Rep. CS-2008-08, Technion-Computer Science Department, 2008.
- [28] R. Deriche, "Recursively implementing the gaussian and its derivatives," in *Proceedings of IEEE International Conference on Image Processing (ICIP '92)*, pp. 263–267, 1992.
- [29] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27th Asilomar Conference on Signals, Systems & Computers*, pp. 40–44, November 1993.
- [30] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.
- [31] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
- [32] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via L^1 minimization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [33] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.
- [34] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: a feature similarity index for image quality assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011.
- [35] P. Carr and R. Hartley, "Improved single image dehazing using geometry," in *Proceedings of the Digital Image Computing: Techniques and Applications (DICTA '09)*, pp. 103–110, December 2009.

Research Article

Fast Total-Variation Image Deconvolution with Adaptive Parameter Estimation via Split Bregman Method

Chuan He,¹ Changhua Hu,¹ Wei Zhang,² Biao Shi,¹ and Xiaoxiang Hu¹

¹ Unit 302, Xi'an Institute of High-tech, Xi'an 710025, China

² Unit 403, Xi'an Institute of High-tech, Xi'an 710025, China

Correspondence should be addressed to Chuan He; hechuan8512@163.com

Received 16 August 2013; Accepted 27 December 2013; Published 17 February 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Chuan He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The total-variation (TV) regularization has been widely used in image restoration domain, due to its attractive edge preservation ability. However, the estimation of the regularization parameter, which balances the TV regularization term and the data-fidelity term, is a difficult problem. In this paper, based on the classical split Bregman method, a new fast algorithm is derived to simultaneously estimate the regularization parameter and to restore the blurred image. In each iteration, the regularization parameter is refreshed conveniently in a closed form according to Morozov's discrepancy principle. Numerical experiments in image deconvolution show that the proposed algorithm outperforms some state-of-the-art methods both in accuracy and in speed.

1. Introduction

Digital image restoration, which aims at recovering an estimate of the original scene from the degraded observation, is a recurrent task with many real-world applications, for example, remote sensing, astronomy, and medical imaging. During acquisition, the observed images are often degraded by relative motion between the camera and the original scene, defocusing of the lens system, atmospheric turbulence, and so forth. In most cases, the degradation can be modeled as a spatially linear shift invariant system, where the original image is convolved by a spatially invariant point spread function (PSF) and contaminated with Gaussian white noise [1].

Without loss of generality, we assume that the digital grayscale images used throughout this paper have an $m \times n$ domain and are represented by mn vectors formed by stacking up the image matrix rows. So the (i, j) th pixel becomes the $((i - 1)n + j)$ th entry of the vector. Then, in general, the degradation process can be modeled as the following discrete linear inverse problem:

$$\mathbf{f} = \mathbf{H}\mathbf{u}_{\text{clean}} + \mathbf{n}, \quad (1)$$

where \mathbf{f} and $\mathbf{u}_{\text{clean}}$ are the observed image and the original image, respectively, both expressed in vectorial form, \mathbf{H} is

the convolution operator in accordance with the spatially invariant PSF, which is assumed to be known, and \mathbf{n} is a vector of zero mean Gaussian white noise of variance σ^2 . In most cases, \mathbf{H} is ill-conditioned so that directly estimating $\mathbf{u}_{\text{clean}}$ from \mathbf{f} is of no possibility. The solution of (1) is highly sensitive to noise in the observed image and it becomes a well-known ill-posed linear inverse problem (IPLIP). The inverse filtering in a least square form, which tries to solve this problem directly, usually results in an estimation of no usability.

If we get some prior knowledge such as prior distribution or sparse quality about the original image, we can incorporate such information into the restoration process via some sort of regularization [2]. This makes the solution of IPLIP possible. A large class of regularization approaches leads to the following minimization problem:

$$\min_{\mathbf{u}} \left\{ \Phi(\mathbf{u}) + \frac{\lambda}{2} \|\mathbf{H}\mathbf{u} - \mathbf{f}\|_2^2 \right\}, \quad (2)$$

where \mathbf{u} is the estimate of $\mathbf{u}_{\text{clean}}$ and λ is the so-called regularization parameter. The first term of (2) represents the regularization term, whereas the second represents the data-fidelity term. The regularization has the quality of numerical stabilizing and encourages the result to have some desirable

properties. The positive regularization parameter λ plays the role of balancing the relative weight of the two terms.

Among the various regularization methods, the total-variation (TV) regularization is famed for its attractive edge preservation ability. It was introduced into image restoration by Rudin et al. [3] in 1992. From then on, the TV regularization has been arousing significant attention [4–7], and, so far, it has resulted in several variants [8–10]. The objective functional of the TV restoration problem is given by

$$\min_{\mathbf{u}} \left\{ \sum_i \|\mathbf{D}_i \mathbf{u}\|_2 + \frac{\lambda}{2} \|\mathbf{H}\mathbf{u} - \mathbf{f}\|_2^2 \right\}, \quad (3)$$

where the first term is the so-called TV seminorm of \mathbf{u} and $\mathbf{D}_i \mathbf{u}$ (its detailed definition is in Section 2) is the discrete gradient of \mathbf{u} at pixel i . In minimization functional (3), the TV is either isotropic if $\|\cdot\|$ is 2-norm or anisotropic if it is 1-norm. We emphasize here that our method is applicable to both isotropic and anisotropic cases. However, we will only treat the isotropic one for simplicity, since the treatment for the other one is completely analogous. Despite the advantage of edge preservation, the minimization of functional (3) is troublesome and it has no closed form solution at all. Various methods have been proposed to minimize (3), including time-marching schemes [3], primal-dual based methods [11–13], fixed point iteration approaches [14], and variable splitting algorithms [15–17]. In particular, the split Bregman method adopted in this paper is an instance of the variable splitting based algorithms.

Another critical issue in TV regularization is the selection of the regularization parameter λ , since it plays a very important role. If λ is too large, the regularized solution will be undersmoothed, and, on the contrary, if λ is too small, the regularized solution will not fit the observation properly. Most works in the literature only consider a fixed λ and, when applying these methods to image restoration problems, one should adjust λ manually to get a satisfying solution. So far, a few strategies are proposed for the adaptive estimation of parameter λ , for example, the L-curve method [18], the variational Bayesian approach [19], the generalized cross-validation (GCV) method [20], and Morozov's discrepancy principle [21].

If the noise level is available or can be estimated first, Morozov's discrepancy principle is a good choice for the selection of λ . According to this rule, the TV image restoration problem can be described as

$$\min_{\mathbf{u}} \sum_i \|\mathbf{D}_i \mathbf{u}\|_2 \quad \text{s.t. } \mathbf{u} \in \mathbf{S}, \quad (4)$$

where $\mathbf{S} := \{\mathbf{u} : \|\mathbf{H}\mathbf{u} - \mathbf{f}\|_2^2 \leq c\}$ with $c = \tau m n \sigma^2$ is the feasible set in accordance with the discrepancy principle. Although it is much easier to solve the unconstrained problem (3) than the constrained problem (4), formulation (4) has a clear physics meaning (c is proportional to the noise variance) and this makes the estimation of λ easier. In fact, referring to the theory of Lagrangian methods, if \mathbf{u} is a solution of constrained problem (4), it will also be a solution of (3) for a particular choice of $\lambda \geq 0$, which is the Lagrangian multiplier

corresponding to the constraint in (4). To minimize (4), we have either $\mathbf{u} \in \mathbf{S}$ for $\lambda = 0$ or

$$\|\mathbf{H}\mathbf{u} - \mathbf{f}\|_2^2 = c \quad (5)$$

for $\lambda > 0$. In fact, if $\lambda = 0$, minimizing (3) is equivalent to minimizing $\sum_i \|\mathbf{D}_i \mathbf{u}\|_2$, which means that the solution is a constant image. Obviously, this will not happen to a nature image. Therefore, only $\lambda > 0$ will happen in practical applications.

There exists no closed form solution of functional (3) or (4), and, up to now, several papers pay attention to the numerical solving of problem (4). In [22], the authors provided a modular solver to update λ for making use of existing methods for the unconstrained problems. Afonso et al. [17] proposed an alternating direction method of multipliers (ADMM) based approach and suggested using Chambolle's dual method [23] to adaptively restore the degraded image. In [13], Wen and Chan proposed a primal-dual based method to solve the constrained problem (4). The minimization problem was transformed into a saddle point problem of the primal-dual model of (4), and then the proximal point method [24] was applied to find the saddle point. When dealing with the updating of λ , they resorted to a Newton's inner iteration. All these methods mentioned above have the same limitation: in order to adaptively update λ , an inner iteration is introduced, and this results in extra computing cost.

In this paper, based on the split Bregman scheme, we propose a fast algorithm to solve the constrained TV restoration problem (4). When referring to the variable splitting technique, we introduce two auxiliary variables to represent $\mathbf{D}\mathbf{u}$ and the TV norm, respectively, and therefore the constrained problem (4) can be solved efficiently with a separable structure without any inner iteration. Differing from the previous works focusing on the adaptive regularization parameter estimation in TV restoration problems, our method involves no inner iteration and adjusts the regularization parameter in a closed form in each iteration. Thus, fast computation speed is achieved. The simulation results in TV restoration problems indicate that our method outperforms some famous methods in accuracy and especially in speed. According to the equivalence of split Bregman method, ADMM, and Douglas-Rachford splitting algorithm under the assumption of linear constraints [25–27], our algorithm can also be seen as an instance of ADMM or Douglas-Rachford splitting algorithm.

In the rest of this paper, the basic notation is presented in Section 2. Section 3 gives the derivation leading to the proposed algorithm and some practical parameter setting strategies. In Section 4, several experiments are reported to demonstrate the effectiveness of our algorithm. Finally, Section 5 draws a short conclusion of this paper.

2. Basic Notation

Let us describe the notation that we will be using throughout this paper. Euclidean space R^m is denoted as \mathbf{P} , whereas Euclidean space $R^{m \times m}$ is denoted as $\mathbf{T} := \mathbf{P} \times \mathbf{P}$. The i th

components of $\mathbf{x} \in \mathbf{P}$ and $\mathbf{y} \in \mathbf{T}$ are denoted as $x_i \in R$ and $\mathbf{y}_i = (y_i^{(1)}, y_i^{(2)})^T \in R^2$, respectively. Define inner products $\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbf{P}} = \sum_i x_i x_i$, $\langle \mathbf{y}, \mathbf{y} \rangle_{\mathbf{T}} = \sum_i \sum_{k=1}^2 y_i^{(k)} y_i^{(k)}$, and norm $\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbf{P}}}$, $\|\mathbf{y}\|_2 = \sqrt{\langle \mathbf{y}, \mathbf{y} \rangle_{\mathbf{T}}}$. For each $\mathbf{u} \in \mathbf{P}$, we define $\mathbf{D}_i \mathbf{u} := [(\mathbf{D}^{(1)} \mathbf{u})_i, (\mathbf{D}^{(2)} \mathbf{u})_i]^T$, with

$$(\mathbf{D}^{(1)} \mathbf{u})_i := \begin{cases} u_{i+n} - u_i, & \text{if } 1 \leq i \leq n(m-1), \\ u_{\text{mod}(i,n)} - u_i, & \text{otherwise,} \end{cases} \quad (6)$$

$$(\mathbf{D}^{(2)} \mathbf{u})_i := \begin{cases} u_{i+1} - u_i, & \text{if } \text{mod}(i,n) \neq 0, \\ u_{i-n+1} - u_i, & \text{otherwise,} \end{cases} \quad (7)$$

where $\mathbf{D}^{(1)}, \mathbf{D}^{(2)} \in R^{mn \times mn}$ are $mn \times mn$ matrices in the vertical and horizontal directions, and obviously it holds that $\mathbf{D}^{(1)} \mathbf{u} \in \mathbf{P}$ and $\mathbf{D}^{(2)} \mathbf{u} \in \mathbf{P}$. $\mathbf{D}_i \in R^{2 \times mn}$ is a tow-row matrix formed by stacking the i th rows of $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ together. Define the global first-order finite difference operator as $\mathbf{D} := [(\mathbf{D}^{(1)})^T, (\mathbf{D}^{(2)})^T]^T \in R^{2mn \times mn}$ and we consider $\mathbf{D}\mathbf{u} \in \mathbf{T}$. From (6) and (7), we see that the periodic boundary condition is assumed here.

Given a convex functional $J(\mathbf{z})$, the subdifferential $\partial J(\mathbf{z}_1)$ of $J(\mathbf{z})$ at \mathbf{z}_1 is defined as

$$\partial J(\mathbf{z}_1) := \{\mathbf{q} \in \mathbf{P} : \langle \mathbf{q}, \mathbf{z} - \mathbf{z}_1 \rangle \leq J(\mathbf{z}) - J(\mathbf{z}_1), \forall \mathbf{z} \in \mathbf{P}\}. \quad (8)$$

And the Bregman distance between \mathbf{z} and \mathbf{z}_1 is defined as

$$D_J^{(\mathbf{z}_1)} = J(\mathbf{z}) - J(\mathbf{z}_1) - \langle \mathbf{q}, \mathbf{z} - \mathbf{z}_1 \rangle. \quad (9)$$

From the definition of Bregman distance, we learn that it is positive all the time.

3. Methodology

3.1. Deduction of the Proposed Algorithm. We refer to the variable splitting technique [28] for liberating the discrete operator $\mathbf{D}_i \mathbf{u}$ out from nondifferentiability and simplifying the regularization parameter's updating. An auxiliary variable $\mathbf{x} \in \mathbf{P}$ is introduced for $\mathbf{H}\mathbf{u}$, and another auxiliary variable $\mathbf{y} \in \mathbf{T}$ is introduced to represent $\mathbf{D}\mathbf{u}$ (or $\mathbf{y}_i \in R^2$ for $\mathbf{D}_i \mathbf{u}$, resp.). Therefore, functional (3) is equivalent to

$$\min_{\mathbf{u}, \mathbf{x}, \mathbf{y}} \left\{ \sum_i \|\mathbf{y}_i\|_2 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{f}\|_2^2 \right\} \quad (10)$$

subject to $\mathbf{H}\mathbf{u} = \mathbf{x}$, $\mathbf{y}_i = \mathbf{D}_i \mathbf{u}$, $i = 1, 2, \dots, mn$.

Define Bregman functional

$$J(\mathbf{u}, \mathbf{x}, \mathbf{y}) = \left\{ \sum_i \|\mathbf{y}_i\|_2 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{f}\|_2^2 \right\}. \quad (11)$$

Then the Bregman distance of $J(\mathbf{u}, \mathbf{x}, \mathbf{y})$ is

$$\begin{aligned} D_J^{(\mathbf{p}_u^k, \mathbf{p}_x^k, \mathbf{p}_y^k)}(\mathbf{u}, \mathbf{x}, \mathbf{y}; \mathbf{u}^k, \mathbf{x}^k, \mathbf{y}^k) &= J(\mathbf{u}, \mathbf{x}, \mathbf{y}) - J(\mathbf{u}^k, \mathbf{x}^k, \mathbf{y}^k) \\ &\quad - \langle \mathbf{p}_u^k, \mathbf{u} - \mathbf{u}^k \rangle - \langle \mathbf{p}_x^k, \mathbf{x} - \mathbf{x}^k \rangle \\ &\quad - \langle \mathbf{p}_y^k, \mathbf{y} - \mathbf{y}^k \rangle. \end{aligned} \quad (12)$$

According to the split Bregman method [16, 29], we obtain the following iterative scheme:

$$\begin{aligned} &(\mathbf{u}^{k+1}, \mathbf{x}^{k+1}, \mathbf{y}^{k+1}) \\ &= \arg \min_{\mathbf{u}, \mathbf{x}, \mathbf{y}} \left\{ D_J^{(\mathbf{p}_u^k, \mathbf{p}_x^k, \mathbf{p}_y^k)}(\mathbf{u}, \mathbf{x}, \mathbf{y}; \mathbf{u}^k, \mathbf{x}^k, \mathbf{y}^k) \right. \\ &\quad \left. + \frac{\beta_1}{2} \|\mathbf{x} - \mathbf{H}\mathbf{u}\|_2^2 + \frac{\beta_2}{2} \|\mathbf{y} - \mathbf{D}\mathbf{u}\|_2^2 \right\}, \end{aligned} \quad (13)$$

$$\mathbf{p}_u^{k+1} = \mathbf{p}_u^k + \beta_1 \mathbf{H}^T (\mathbf{x}^{k+1} - \mathbf{H}\mathbf{u}^{k+1}) + \beta_2 \mathbf{D}^T (\mathbf{y}^{k+1} - \mathbf{D}\mathbf{u}^{k+1}), \quad (14)$$

$$\mathbf{p}_x^{k+1} = \mathbf{p}_x^k + \beta_1 (\mathbf{H}\mathbf{u}^{k+1} - \mathbf{x}^{k+1}), \quad (15)$$

$$\mathbf{p}_y^{k+1} = \mathbf{p}_y^k + \beta_2 (\mathbf{D}\mathbf{u}^{k+1} - \mathbf{y}^{k+1}), \quad (16)$$

if we define that

$$\begin{aligned} \mathbf{p}_u^0 &:= -\beta_1 \mathbf{H}^T \mathbf{b}^0 - \beta_2 \mathbf{D}^T \mathbf{d}^0 \\ \mathbf{p}_x^0 &:= \beta_1 \mathbf{b}^0 \\ \mathbf{p}_y^0 &:= \beta_2 \mathbf{d}^0, \end{aligned} \quad (17)$$

for any elements $\mathbf{b}^0 \in \mathbf{P}$ and $\mathbf{d}^0 \in \mathbf{T}$, and then, according to (14)–(16), it holds that

$$\begin{aligned} \mathbf{p}_u^k &= -\beta_1 \mathbf{H}^T \mathbf{b}^k - \beta_2 \mathbf{D}^T \mathbf{d}^k & \mathbf{p}_x^k &= \beta_1 \mathbf{b}^k & \mathbf{p}_y^k &= \beta_2 \mathbf{d}^k \\ & & & & k &= 0, 1, \dots \end{aligned} \quad (18)$$

and we obtain the following iterative scheme:

$$\begin{aligned} &(\mathbf{u}^{k+1}, \mathbf{x}^{k+1}, \mathbf{y}^{k+1}) \\ &= \arg \min_{\mathbf{u}, \mathbf{x}, \mathbf{y}} \left\{ \frac{\lambda}{2} \|\mathbf{x} - \mathbf{f}\|_2^2 + \frac{\beta_1}{2} \|\mathbf{x} - \mathbf{H}\mathbf{u} - \mathbf{b}^k\|_2^2 \right. \\ &\quad \left. + \sum_i \|\mathbf{y}_i\|_2 + \frac{\beta_2}{2} \|\mathbf{y} - \mathbf{D}\mathbf{u} - \mathbf{d}^k\|_2^2 \right\}, \quad (19) \\ &\mathbf{b}^{k+1} = \mathbf{b}^k + \mathbf{H}\mathbf{u}^{k+1} - \mathbf{x}^{k+1}, \\ &\mathbf{d}^{k+1} = \mathbf{d}^k + \mathbf{D}\mathbf{u}^{k+1} - \mathbf{y}^{k+1}. \end{aligned}$$

In iterative scheme (19), the problem yielding $(\mathbf{u}^{k+1}, \mathbf{x}^{k+1}, \mathbf{y}^{k+1})$ exactly is difficult, since it needs an inner iterative scheme. Here, we adopt the alternating direction method (ADM) to approximately calculate \mathbf{u}^{k+1} , \mathbf{x}^{k+1} , and \mathbf{y}^{k+1} in each iteration and this leads to the following iterative framework:

$$\mathbf{u}^{k+1} = \arg \min_{\mathbf{u}} \left\{ \frac{\beta_1}{2} \|\mathbf{x}^k - \mathbf{H}\mathbf{u} - \mathbf{b}^k\|_2^2 + \frac{\beta_2}{2} \|\mathbf{y}^k - \mathbf{D}\mathbf{u} - \mathbf{d}^k\|_2^2 \right\}, \quad (20)$$

$$\mathbf{y}^{k+1} = \arg \min_{\mathbf{y}} \left\{ \sum_i \|\mathbf{y}_i\|_2 + \frac{\beta_2}{2} \|\mathbf{y} - \mathbf{D}\mathbf{u} - \mathbf{d}^k\|_2^2 \right\}, \quad (21)$$

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{x}} \left\{ \frac{\lambda^{k+1}}{2} \|\mathbf{x} - \mathbf{f}\|_2^2 + \frac{\beta_1}{2} \|\mathbf{x} - \mathbf{H}\mathbf{u}^{k+1} - \mathbf{b}^k\|_2^2 \right\}, \quad (22)$$

$$\mathbf{b}^{k+1} = \mathbf{b}^k + \mathbf{H}\mathbf{u}^{k+1} - \mathbf{x}^{k+1}, \quad (23)$$

$$\mathbf{d}^{k+1} = \mathbf{d}^k + \mathbf{D}\mathbf{u}^{k+1} - \mathbf{y}^{k+1}. \quad (24)$$

In the following, we will discuss how to solve problems (20)–(22) efficiently.

The minimization subproblem with respect to \mathbf{u} is in the form of least square. From functional (20), we obtain

$$\left(\frac{\beta_1}{\beta_2} \mathbf{H}^T \mathbf{H} + \mathbf{D}^T \mathbf{D} \right) \mathbf{u} = \frac{\beta_1}{\beta_2} \mathbf{H}^T (\mathbf{x}^k - \mathbf{b}^k) + \mathbf{D}^T (\mathbf{y}^k - \mathbf{d}^k). \quad (25)$$

Under the periodic boundary condition, matrices \mathbf{H} , $\mathbf{D}^{(1)}$, and $\mathbf{D}^{(2)}$ are block-circulant, so they can be diagonalized by a Discrete Fourier Transforms (DFTs) matrix. Using the convolution theorem of Fourier Transforms, we obtain

$$\begin{aligned} \mathbf{u}^{k+1} = \mathcal{F}^{-1} & \left(\left(\frac{\beta_1}{\beta_2} \right) \mathcal{F}^* (\mathbf{H}) \circ \mathcal{F} (\mathbf{x}^k - \mathbf{b}^k) \right. \\ & + \mathcal{F}^* (\mathbf{D}^{(1)}) \mathcal{F} \left((\mathbf{y}^k)^{(1)} - (\mathbf{d}^k)^{(1)} \right) \\ & + \mathcal{F}^* (\mathbf{D}^{(2)}) \mathcal{F} \left((\mathbf{y}^k)^{(2)} - (\mathbf{d}^k)^{(2)} \right) \\ & \circ \left(\left(\frac{\beta_1}{\beta_2} \right) \mathcal{F}^* (\mathbf{H}) \circ \mathcal{F} (\mathbf{H}) + \mathcal{F}^* (\mathbf{D}^{(1)}) \right. \\ & \left. \left. \circ \mathcal{F} (\mathbf{D}^{(1)}) + \mathcal{F}^* (\mathbf{D}^{(2)}) \circ \mathcal{F} (\mathbf{D}^{(2)}) \right)^{-1} \right), \end{aligned} \quad (26)$$

where \mathcal{F} denotes the DFT, “ $*$ ” denotes complex conjugate, and “ \circ ” represents componentwise multiplication. The reciprocal notation is also componentwise here. Therefore, problem (20) can be solved by two Fast Fourier Transforms (FFTs) and one inverse FFT in $O(mn \log(mn))$ operations.

Functional (21) is a proximal minimization problem and it can be solved componentwise by a two-dimension shrinkage as follows:

$$\mathbf{y}_i^{k+1} = \max \left\{ \left\| \mathbf{D}_i \mathbf{u}^{k+1} + \mathbf{d}_i^k \right\|_2 - \frac{1}{\beta_2}, 0 \right\} \frac{\mathbf{D}_i \mathbf{u}^{k+1} + \mathbf{d}_i^k}{\left\| \mathbf{D}_i \mathbf{u}^{k+1} + \mathbf{d}_i^k \right\|_2}. \quad (27)$$

During the calculation, we employ the convention $0 \times (0/0) = 0$ to avoid getting results of no meaning.

When dealing with problem (22), we assume that $\mathbf{w}^{k+1} = \mathbf{H}\mathbf{u}^{k+1} + \mathbf{b}^k$ first. It is obvious that \mathbf{x} is λ related and it plays the role of $\mathbf{H}\mathbf{u}$. Therefore, in each iteration, we should examine whether $\|\mathbf{x} - \mathbf{f}\|_2^2 \leq c$ holds true, that is, whether \mathbf{x} meets the discrepancy principle.

The solutions of λ and \mathbf{x} fall into two cases according to the range of \mathbf{w}^{k+1} .

(1) If

$$\left\| \mathbf{w}^{k+1} - \mathbf{f} \right\|_2^2 \leq c \quad (28)$$

holds true, we set $\lambda^{k+1} = 0$ and $\mathbf{x}^{k+1} = \mathbf{w}^{k+1}$. Obviously this \mathbf{x}^{k+1} satisfies the discrepancy principle.

(2) If $\left\| \mathbf{w}^{k+1} - \mathbf{f} \right\|_2^2 > c$, according to the discrepancy principle, we should solve equation

$$\left\| \mathbf{w}^{k+1} - \mathbf{f} \right\|_2^2 = c. \quad (29)$$

Since the minimization problem (22) with respect to \mathbf{x} is quadratic, it has a closed form solution

$$\mathbf{x}^{k+1} = \frac{(\lambda^{k+1} \mathbf{f} + \beta_1 \mathbf{w}^{k+1})}{(\lambda^{k+1} + \beta_1)}. \quad (30)$$

Substituting \mathbf{x}^{k+1} in (29) with (30), we obtain

$$\lambda^{k+1} = \frac{\beta_1 \left\| \mathbf{f} - \mathbf{w}^{k+1} \right\|_2}{\sqrt{c}} - \beta_1. \quad (31)$$

The above discussion can be summed up by Algorithm 1.

In algorithm APE-SBA, by introducing the auxiliary variable \mathbf{x} , $\mathbf{H}\mathbf{u}$ is liberated out from the constraint of the discrepancy principle, and consequently a closed form to update λ is obtained without any inner iteration. This is the major difference between APE-SBA and the methods in [13] and [17]. Since the procedure of solving (26) corresponding to the \mathbf{u} subproblem consumes the most, the calculation cost of our algorithm is about $O(mn \log(mn))$ FFT operations. In fact, our algorithm is an instance of the classical split Bregman method, so the convergence of it is guaranteed by the theorem proposed by Eckstein and Bertsekas [30]. We summarize the convergence of our algorithm as follows.

Theorem 1. For $\beta_1, \beta_2 > 0$, the sequence $\{\mathbf{u}^k, \mathbf{x}^k, \mathbf{y}^k, \mathbf{b}^k, \mathbf{d}^k, \lambda^k\}$ generated by Algorithm APE-SBA from any initial point $(\mathbf{u}^0, \mathbf{x}^0, \mathbf{b}^0, \mathbf{d}^0)$ converges to $(\mathbf{u}^*, \mathbf{x}^*, \mathbf{y}^*, \mathbf{b}^*, \mathbf{d}^*, \lambda^*)$, where $(\mathbf{u}^*, \mathbf{x}^*, \mathbf{y}^*)$ is a solution of the functional (10). In particular, \mathbf{u}^* is the minimizer of functional (4), and λ^* is the Lagrange multiplier corresponding to constraint $\mathbf{u} \in \mathbf{S}$ according to the unconstrained problem (3).

3.2. Parameter Setting. In this paper, the noise level is denoted by the following defined blurred signal-to-noise ratio (BSNR)

$$\text{BSNR} = 10 \log_{10} \left(\frac{\left\| \mathbf{f} - \bar{\mathbf{f}} \right\|_2^2}{mn\sigma^2} \right), \quad (32)$$

where $\bar{\mathbf{f}}$ denotes the mean of \mathbf{f} .

In minimization problem (4), the noise dependent upper bound c is very important, since a good choice of it can constrain the error between the restored image and the original image to a reasonable level. To our best knowledge, the choice of this parameter is an open problem which has not been solved theoretically. One approach to choose c is referring to the equivalent degrees of freedom (DF), but the calculation of DF is a difficult problem and we can only get

Input: $\mathbf{f}, \mathbf{H}, c$.

- (1) Initialize $\mathbf{u}^0, \mathbf{x}^0, \mathbf{b}^0, \mathbf{d}^0$. Set $k = 0$ and $\beta_1 > 0$ and $\beta_2 > 0$
- (2) **while** stopping criterion is not satisfied, **do**
- (3) Compute \mathbf{u}^{k+1} according to (26);
- (4) Compute \mathbf{y}^{k+1} according to (27);
- (5) **if** (28) holds, then
- (6) $\lambda^{k+1} = 0$, and $\mathbf{x}^{k+1} = \mathbf{w}^{k+1}$;
- (7) **else**
- (8) Update λ^{k+1} and \mathbf{x}^{k+1} according to (31) and (30);
- (9) **end if**
- (10) Update \mathbf{b}^{k+1} and \mathbf{d}^{k+1} according to (23) and (24);
- (11) $k = k + 1$;
- (12) **end while**
- (13) **return** λ^{k+1} and \mathbf{u}^{k+1} .

ALGORITHM 1: APE-SBA: Adaptive Parameter Estimation Split Bregman Algorithm.

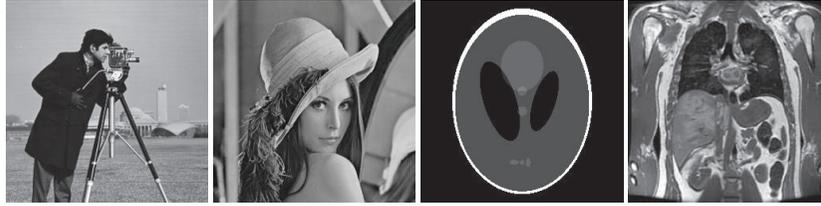


FIGURE 1: Test images: *Cameraman*, *Lena*, *Shepp-Logan phantom*, and *Abdomen* of size 256×256 .

an estimate of it. A simple strategy of choosing c is to employ a curve approximating the relation between the noise level and τ . By fitting experimental data with a straight line, in this paper, we suggest setting

$$\tau = -0.006 \times \text{BSNR} + 1.09. \quad (33)$$

Besides the setting of τ , the choice of β_1 and β_2 is essential to our algorithm. We suggest setting $\beta_1 = 10^{(\text{BSNR}/10-1)} \times \beta_2$, where $\beta_2 = 1$. This parameter setting is obtained by large numbers of experiments. Actually, $\beta_1, \beta_2 > 0$ is sufficient for the convergence of the proposed algorithm, but why β_1 and β_2 play different important role when the BSNR varies? The reason is that, when the BSNR becomes higher, the distance between \mathbf{Hu} and \mathbf{f} is nearer. From minimization problem (10), we learn that auxiliary variable \mathbf{x} plays the role of \mathbf{Hu} and a higher BSNR means a larger β_1 .

4. Numerical Results

In this section, two experiments are presented to demonstrate the effectiveness of the proposed method. They were performed under MATLAB v7.8.0 and Windows 7 on a PC with Intel Core (TM) i5 CUP (3.20 GHz) and 8 GB of RAM. The improved signal-to-noise ratio (ISNR) is used to measure the quality of the restoration results. It is defined as

$$\text{ISNR} = 10 \log_{10} \left(\frac{\|\mathbf{f} - \mathbf{u}_{\text{clean}}\|_2^2}{\|\mathbf{u} - \mathbf{u}_{\text{clean}}\|_2^2} \right). \quad (34)$$

During the experiments, the four images shown in Figure 1 were used; they are named *Cameraman*, *Lena*, *Shepp-Logan phantom*, and *Abdomen* all of size 256×256 .

4.1. Experiment 1. In this experiment, we examine whether the regularization parameter is well estimated by the proposed algorithm. We compare APE-SBA with some famous TV-based methods in the literature and they are denoted by BFO [5], BMK [19], and LLN [20]. We make use of MATLAB commands “fspecial (“average”, 9)” and “fspecial (“Gaussian”, [9 9], 3)” to blur the *Lena*, *Cameraman*, and *Shepp-Logan phantom* images first, and then the images are contaminated with Gaussian noises such that the BSNRs of the observed images are 20 dB, 30 dB, and 40 dB. We adopt $\|\mathbf{u}^{k+1} - \mathbf{u}^k\|_2^2 / \|\mathbf{u}^k\|_2^2 \leq 10^{-6}$ as the stopping criteria for our algorithm, where \mathbf{u}^k is the restored image in the k th iteration.

Table 1 presents the ISNRs of the restoration results of different methods. Symbol “—” means that the results are not presented in the original reference, and bold type numbers represent the best results among the four methods. From Table 1, we see that our algorithm is more competitive than the other three and only in one case our result is worse than but close to the best. This also indicates that the regularization parameter obtained by our method is good.

4.2. Experiment 2. In this subsection, we compare our algorithm with the other two state-of-the-art algorithms: the primal-dual based method in [13], named AutoRegSel, and the ADMM based method in [17], named C-SALSA. The

TABLE 1: ISNRs obtained by different methods.

BSNR	Method	Lena	Cameraman	Shepp-Logan
		ISNR (dB)	ISNR (dB)	ISNR (dB)
9 × 9 uniform blur				
20	BFO [5]	4.05	3.27	6.25
	BMK [19]	3.72	2.42	3.01
	LLN [20]	3.15	2.88	—
30	APE-SBA	4.09	3.88	7.60
	BFO [5]	5.43	5.69	10.49
	BMK [19]	5.89	5.41	7.77
40	LLN [20]	4.43	5.57	—
	APE-SBA	5.97	5.87	11.56
	BFO [5]	6.22	8.46	16.39
40	BMK [19]	8.42	8.57	13.69
	LLN [20]	6.92	7.86	—
	APE-SBA	8.11	8.60	17.80
9 × 9 Gaussian blur				
20	BFO [5]	2.99	2.21	4.24
	BMK [19]	2.87	1.72	1.85
	LLN [20]	2.57	1.82	—
30	APE-SBA	3.10	2.61	5.92
	BFO [5]	3.82	3.59	7.21
	BMK [19]	3.87	2.63	4.31
40	LLN [20]	4.17	3.43	—
	APE-SBA	4.20	4.17	8.87
	BFO [5]	4.41	5.78	10.27
40	BMK [19]	4.78	3.39	6.69
	LLN [20]	5.44	5.02	—
	APE-SBA	5.97	6.38	11.08

TABLE 2: Comparison between different methods in terms of ISNR, iterations, and runtime.

Problem	Method	Abdomen 256 × 256			Lena 256 × 256		
		ISNR (dB)	Iterations	Runtime (s)	ISNR (dB)	Iterations	Runtime (s)
Prob. 1	APE-SBA	9.63	261	2.31	7.44	201	1.78
	AutoRegSel [13]	9.41	435	6.53	7.39	392	5.94
	C-SALSA [17]	9.14	773	20.44	6.98	658	17.42
Prob. 2	APE-SBA	5.54	551	4.90	4.08	421	3.75
	AutoRegSel [13]	5.24	855	12.92	3.96	1000	15.52
	C-SALSA [17]	5.00	533	14.03	3.66	492	12.89
Prob. 3	APE-SBA	8.87	263	2.34	6.78	197	1.75
	AutoRegSel [13]	8.54	414	6.26	6.65	404	6.11
	C-SALSA [17]	8.20	422	11.08	6.20	507	13.29

stopping criterion of all methods is $\|\mathbf{u}^{k+1} - \mathbf{u}^k\|_2^2 / \|\mathbf{u}^k\|_2^2 \leq 10^{-6}$ or the number of iterations is larger than 1000. We consider the three image restoration problems adopted in [17]. In the first problem, the PSF is a 9×9 uniform blur with noise variance 0.56^2 (Prob. 1); in the second problem, the PSF is a 9×9 Gaussian blur with noise variance 2 (Prob. 2); in the third problem, the PSF is given by $h_{i,j} = 1/(1 + i^2 + j^2)$ with noise variance 2 (Prob. 3), where $i, j = -7, \dots, 7$.

The plots of ISNR (in dB) versus runtime (in second) are shown in Figure 2. Table 2 presents the ISNR values, the number of iterations, and the total runtime to reach

convergence. We again use the bold type numbers to represent the best results. From the results, we see that APE-SBA produces the best ISNRs compared with the other methods within the least runtime. Besides, in most cases, APE-SBA obtains the best ISNR within the least iterations. Only when dealing with the *Abdomen* image under Prob. 2, APE-SBA takes more iterations but less runtime to reach convergence than C-SALSA, and the total iteration number for these two is close to each other. For achieving the adaptive image restoration, both C-SALSA and AutoRegSel introduce in an inner iterative scheme, whereas APE-SBA contains no

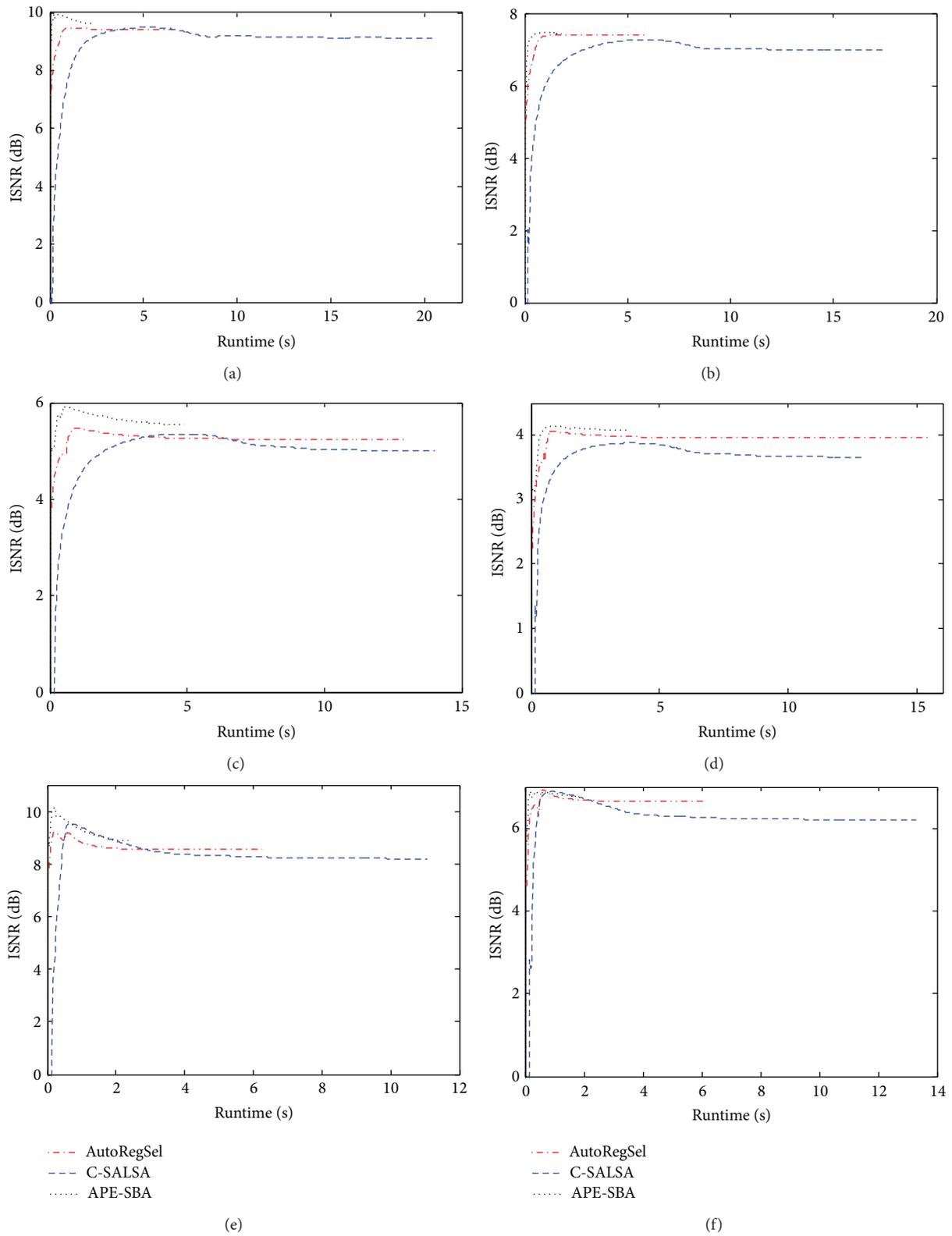


FIGURE 2: ISNR versus runtime for the (left) *Abdomen* image and (right) *Lena* image, which are blurred by a 9×9 uniform blur with noise variance 0.56^2 (first row), by a 9×9 Gaussian blur with noise variance 2 (second row), and by PSF given by $h_{ij} = 1/(1+i^2+j^2)$ ($i, j = -7, \dots, 7$) with noise variance 2 (third row).

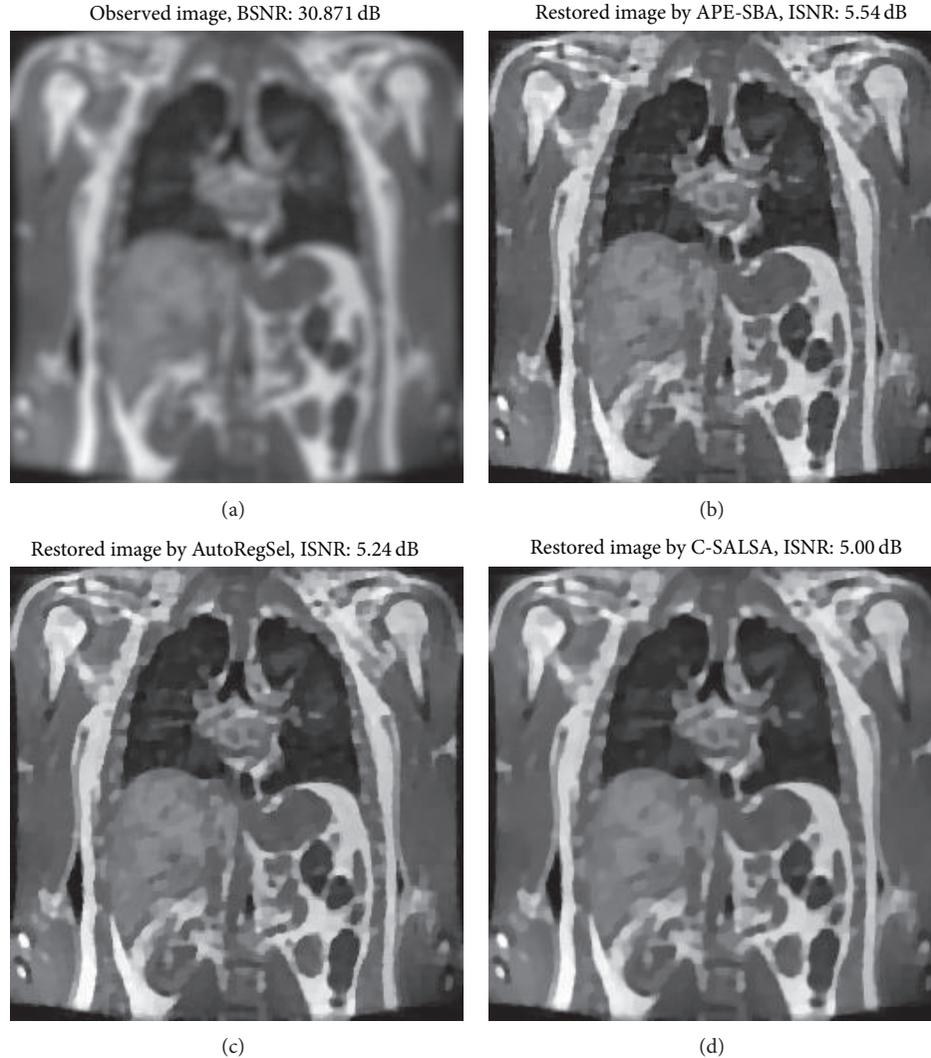


FIGURE 3: The observed image (a) which is degraded by a 9×9 Gaussian blur with noise variance 2, and the restored images by APE-SBA (b), by AutoRegSel (c), and by C-SALSA (d) of the *Abdomen* image under Prob. 2.

inner iteration. Obviously, the superiority in speed of our method will be enlarged when the image size becomes larger. Figure 3 shows the blurred image and the restored results by different methods in Prob. 2 of the *Abdomen* image. Our algorithm results in the best ISNR, and, for other problems in Experiment 2, we obtain the similar results.

5. Conclusions

We developed a split Bregman based algorithm to solve the TV image restoration/deconvolution problem. Unlike some other methods in the literature, without any inner iteration, our method achieves the updating of the regularization parameter and the restoration of the blurred image simultaneously, by referring to the operator splitting technique and introducing two auxiliary variables for both the data-fidelity term and the TV regularization term. Therefore, the algorithm can run without any manual interference. The numerical results have indicated that the proposed algorithm

outperforms some state-of-the-art methods in both speed and accuracy.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61203189 and 61304001 and the National Science Fund for Distinguished Young Scholars of China under Grant 61025014.

References

- [1] H. Andrew and B. Hunt, *Digital Image Restoration*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1977.

- [2] C. R. Vogel, *Computational Methods for Inverse Problems*, vol. 23, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2002.
- [3] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, 1992.
- [4] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Recent developments in total variation image restoration," in *Mathematical Models of Computer Vision*, Springer, New York, NY, USA, 2005.
- [5] J. M. Bioucas-Dias, M. A. T. Figueiredo, and J. P. Oliveira, "Adaptive total variation image deblurring: a majorization-minimization approach," in *Proceedings of the European Signal Processing Conference (EUSIPCO '06)*, Florence, Italy, 2006.
- [6] W. K. Allard, "Total variation regularization for image denoising. III. Examples," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 532–568, 2009.
- [7] W. Stefan, R. A. Renaut, and A. Gelb, "Improved total variation-type regularization using higher order edge detectors," *SIAM Journal on Imaging Sciences*, vol. 3, no. 2, pp. 232–251, 2010.
- [8] Y. Hu and M. Jacob, "Higher degree total variation (HDTV) regularization for image recovery," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2559–2571, 2012.
- [9] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.
- [10] K. Bredies, Y. Dong, and M. Hintermüller, "Spatially dependent regularization parameter selection in total generalized variation models for image restoration," *International Journal of Computer Mathematics*, vol. 90, no. 1, pp. 109–123, 2013.
- [11] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM Journal on Scientific Computing*, vol. 20, no. 6, pp. 1964–1977, 1999.
- [12] B. He and X. Yuan, "Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective," *SIAM Journal on Imaging Sciences*, vol. 5, no. 1, pp. 119–149, 2012.
- [13] Y.-W. Wen and R. H. Chan, "Parameter selection for total-variation-based image restoration using discrepancy principle," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1770–1781, 2012.
- [14] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM Journal on Scientific Computing*, vol. 17, no. 1, pp. 227–238, 1996.
- [15] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 248–272, 2008.
- [16] T. Goldstein and S. Osher, "The split Bregman method for L_1 -regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.
- [17] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 681–695, 2011.
- [18] H. W. Engl and W. Grever, "Using the L -curve for determining optimal regularization parameters," *Numerische Mathematik*, vol. 69, no. 1, pp. 25–31, 1994.
- [19] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational Bayesian blind deconvolution using a total variation prior," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 12–26, 2009.
- [20] H. Liao, F. Li, and M. K. Ng, "Selection of regularization parameter in total variation image restoration," *Journal of the Optical Society of America A*, vol. 26, no. 11, pp. 2311–2320, 2009.
- [21] V. A. Morozov, *Methods for Solving Incorrectly Posed Problems*, Springer, New York, NY, USA, 1984, translated from the Russian by A. B. Aries, translation edited by Z. Nashed.
- [22] P. Blomgren and T. F. Chan, "Modular solvers for image restoration problems using the discrepancy principle," *Numerical Linear Algebra with Applications*, vol. 9, no. 5, pp. 347–358, 2002.
- [23] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1–2, pp. 89–97, 2004, Special issue on mathematics and image analysis.
- [24] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, NY, USA, 2011.
- [25] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for L_1 -minimization with applications to compressed sensing," *SIAM Journal on Imaging Sciences*, vol. 1, no. 1, pp. 143–168, 2008.
- [26] C. Wu and X.-C. Tai, "Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 300–339, 2010.
- [27] S. Setzer, "Operator splittings, Bregman methods and frame shrinkage in image processing," *International Journal of Computer Vision*, vol. 92, no. 3, pp. 265–280, 2011.
- [28] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, vol. 9 of *SIAM Studies in Applied Mathematics*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1989.
- [29] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005.
- [30] J. Eckstein and D. Bertsekas, "On the Douglas Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

Research Article

A Fast Iterative Pursuit Algorithm in Robust Face Recognition Based on Sparse Representation

Zhao Jian,¹ Huang Luxi,¹ Jia Jian,² and Xie Yu¹

¹ School of Information Science and Technology, Northwest University, Xi'an 710069, China

² Department of Mathematics, Northwest University, Xi'an 710069, China

Correspondence should be addressed to Zhao Jian; zjctec@nwu.edu.cn

Received 3 October 2013; Revised 19 December 2013; Accepted 28 December 2013; Published 13 February 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Zhao Jian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A relatively fast pursuit algorithm in face recognition is proposed, compared to existing pursuit algorithms. More stopping rules have been put forward to solve the problem of slow response of OMP, which can fully develop the superiority of pursuit algorithm—avoiding to process useless information in the training dictionary. For the test samples that are affected by partial occlusion, corruption, and facial disguise, recognition rates of most algorithms fall rapidly. The robust version of this algorithm can identify these samples automatically and process them accordingly. The recognition rates on ORL database, Yale database, and FERET database are 95.5%, 93.87%, and 92.29%, respectively. The recognition performance under various levels of occlusion and corruption is also experimentally proved to be significantly enhanced.

1. Introduction

Face recognition is one of the most active and challenging subject in computer vision and artificial intelligence, which has a wide range of applications such as personnel sign system, image search engine, and convicts detecting system. It has been experimentally proved that various sparse representation methods perform well in face recognition [1]. Given sufficient face images of k object members and a test image y , which belongs to one of the object classes, the problem of face recognition can be transformed into a classification issue [1, 2]. The basic idea of this kind of algorithms is to find a sparsest solution to represent y for classification [3, 4].

But current algorithms based on sparse representation also have drawbacks. On one hand, convergence speeds of these methods are slow to some extent. On the other hand, when the test image is under large percent of random corruption or contiguous occlusion, the sparse solution becomes denser so that it is hard for the system to find the right class y belongs to [5–7].

In this paper, a fast pursuit algorithm is proposed to solve the problem mentioned above. A common problem of pursuit algorithms is that the computational speed is quite slow. Aiming at face recognition, we improved the greedy algorithm,

making it much faster than kindred ones. By focusing on crucial information to classification issue—sparsity of errors, this algorithm enhances the robustness in the problem of face misalignment and large percent of occlusion. The basic frame of this paper is as follows: to begin with, we briefly review existing techniques for face recognition, including its advantages and brittleness to occlusion. Then, an improved algorithm is proposed and its feasibility and effectiveness will be demonstrated. Finally, we present experiments on ORL, Yale, and FERET face databases, as well as on a face database collected by ourselves, to verify the modified algorithm.

2. A Review of Pursuit Algorithms to Solve Face Recognition Issue

Pursuit algorithms are for solving the problem (p_0)

$$(p_0) : \min_x \|x\|_0 \quad \text{subject to } y = Ax, \quad (1)$$

where y represents the test sample, A is the training dictionary of the k known classes, and x is the sparse solution [8–10]. The core idea of these greedy algorithms is to update support and provisional solution iteratively in order to reduce the residual to a minimum [11]. Suppose the matrix A is

composed of n training face images of the k subjects, where an image is represented by a column of $A(m * n)$,

$$A = [A_1, A_2, \dots, A_k] = [v_{1,1}, v_{1,2}, \dots, v_{k,n_k}], \quad (2)$$

where A_i represents the i th class of A and $v_{i,j}$ represents j th element in A_i . And the given test sample y , well aligned, can be seen as one of the linear combination of these n -dimensional column vectors:

$$y = \sum_{i=1}^n a_{1,i}v_{1,1} + a_{2,i}v_{1,2} + \dots + a_{k,i}v_{k,n_k}, \quad (3)$$

where $a_{i,j}$ is the corresponding coefficient of v_{i,n_i} . The task of these algorithms is to find the sparsest solution so that class of the largest coefficient can be considered of the same class as y [12].

Now, let us briefly review the procedure of OMP and MP: firstly, find the i th column which can minimize $\varepsilon(i) = \|a_i v_i - y\|_2$, add this column to support, and compute a_i to make the residual minimized. Then, continue to update support and provisional solution iteratively until the residual is less than a threshold we set before. As to MP, which is similar to OMP, the apparent difference is that the coefficients of S^{k-1} original entries remain unchanged, rather than solving a least square for reevaluating all the update support stages.

The greedy strategy expands the support set, initially empty, by one additional column. So the enumeration takes n^{k_0} steps if the optimization problem is known to have k_0 nonzero members, which seems quite slow in many situations. But to the problem of face recognition, since the test sample y belongs to one class in A , in another word, correlated with only a few columns in A , k_0 is a controllable whose value is small. Many iterative-shrinkage algorithms, which “shrink” entry of every column in A to update the optimal solution iteratively in order to make the solution become “sparser,” additionally process $(n - 1)$ classes of useless information every stage. Compared to them, pursuit algorithms can find the most possible class of test image y by the foremost iterations. Just because of this extraordinary nature, we hold that the pursuit method is best for classification via sparse representation.

3. A Fast and Robust Pursuit Algorithm

One of the apparent drawbacks of pursuit algorithm is that the computing speed is slow in many situations, since it must continuously update the support and optimal solution until the result is accurate enough. In this section, we propose an improved pursuit algorithm for face recognition, which mainly involved two aspects: (1) general stopping rules to make the greedy algorithm faster in face recognition; (2) stopping rules for well-aligned and noiseless input samples.

Most state-of-the-art algorithms dealing with sparse representation try to find linear combination of matrix A , which approximates to input vector y . Goal of many kinds of iterative shrinkage algorithms is to “shrink” the sparse solution [13], in other words, to make the solution become “sparser” [14]; on the contrary, pursuit strategy is to make

the solution from sparse to dense. It is known that the basic idea of pursuit algorithm is to add columns to the support and update provisional solution until the residual between the proposed solution and the input vector y is small enough. We can easily imagine that it is hard to represent y by the linear combination of columns in A accurately when the test sample y is occluded or corrupted, as the distance between y and the corresponding column in A increases, and y may be relevant with more columns in A [15].

3.1. General Stopping Rules. Since test image y is related with just a few columns in matrix A , which belong to the same class, only coefficients of one class are valid to the recognition result. So it is unnecessary and inefficient to recover y precisely, which will unavoidable involve many irrelevant classes. Rather than recovering y accurately, our goal is to find the right class rapidly in face recognition; therefore more stopping rules are needed to make the algorithm faster.

3.1.1. Maximum Iterating Times. Let us consider the best situation first. If the test image y is the same as one column in matrix A , indicating that y and a_i are parallel, only one iteration which needs n flops to find that the maximum $(a_i^T y)^2$ is required to identify the class. When the test image is under random corruption or varying level of contiguous occlusion, the errors between y and every column of A become larger. Then y may be correlated with more classes in A . But the final recognition result depends on the largest coefficients of sparse solution, which suggest the most possible class of y . Hence even under the worst condition, n iterating times are enough because the iteration can be seen as an ergodic one to a classification problem where only one class is valid.

3.1.2. Results Resemblance of Two Successive Iterations. As what we have discussed above, our final identification result depends on the largest entries of the sparse solution; therefore we can neglect the details of the solution [6]. The minor change of the solution x may affect the accuracy of representation, but this would not influence the classification result. So if $\|x_k - x_{k-1}\|_2$ is smaller than some predetermined threshold, the iteration process can be stopped.

3.2. Stopping Rules and Processing Aiming at Different Image Samples. Under most conditions, we would not hope that the iteration times reach its upper limit: on one hand, it may take quite a long time; on the other, since the solution becomes denser when iteration times increase, it is harder to identify the right class. And the basic stopping rule of OMP, $\|r\|_2 < \varepsilon_0$, where $r = y - Ax$ and ε_0 is the error threshold, is often hard to obtain as test image may contain various noises or be disguised. Hence, more stopping rules should be devised to reduce iteration times. Now, we discuss this issue in two contexts—well-aligned, noiseless input samples and corrupted or occluded images.

3.2.1. Maximum Coefficient of the Sparse Solution. Suppose the input image contains little noise and is well-aligned, it is easy for the system to classify this sample. What we should do

is to raise the identification speed. The coefficients of sparse solution

$$x_i = \frac{a_i^T r}{\|a_i\|_2}, \quad \text{where } r = y - Ax^{k-1}, \quad (4)$$

x_i reflects the degree of resemblance between test sample y and one column in A — a_i . If the maximum coefficient is large enough, in other words, approximating to 1, we can consider that y belongs to the class of x_i , since y and x_i have already been sufficiently similar and we do not need to represent the minor errors in the linear combination of other columns as the small coefficients are senseless to classification and this process will surely increase the iteration times.

3.2.2. Sparse Level of the Sparse Solution. To corrupted or disguised test samples, however, large coefficients of the sparse solution are hard to reach owing to relatively larger range of noise. But the special characteristic of recognition or identification ensures that the recognition system can find the class of y before the representation error reaches the upper bound we have set. One of the key points is the sparse level of the sparse solution. We define the sparse level as the ratio of two largest coefficients which are in different classes,

$$k = \frac{a_i}{a_j}, \quad (5)$$

where a_i is the largest coefficient, which belongs to class i and a_j is the second largest coefficient whose class is different from class i .

3.2.3. The Noise Level. We define $\mu(A)$ as the maximal inner product between columns in A and test sample y ,

$$\mu(A) = \max [a_i^T \cdot y] \quad (6)$$

which reflects the maximum correlation between y and face images in training dictionary A . If $\mu(A)$ is smaller than some predefined threshold, it means that similarity between y and each column of A is less than that threshold, which indicates that the input sample is not a valid one or with much additional noise. Suppose the test sample is a valid one, it is inefficient and laborious for the system to process y directly since our goal is to try to find the one resemble y in the dictionary, but y itself is not precise enough. Both the random noise and the part of occlusion can be regarded as noise in the test image accordantly. So $\mu(A)$ implies the noise level of the input image.

3.2.4. Stopping Rules for Well-Aligned and Input Samples with Little Noise. The entries of sparse solution x reflect how much their respective contribution to the solution in respect that x includes all coefficients of the combination of columns in A . Matching pursuit algorithm update support and solution based on minimizing the errors

$$\varepsilon(i) = \min \|a_i x_i - r^{k-1}\|_2^2, \quad (7)$$

where $r = y - Ax^{k-1}$ and $x_i = a_i^T r^{k-1} / \|a_i\|_2^2$. For input samples with noise or misaligned, obviously, the coefficients of x decrease. So the largest entry threshold cannot be reached. But if the corrupted percent of the input image y is not too large, the corresponding level of y and a_i , which is in the same class as y , is relatively large compared to the others. This means that the solution is still quite sparse after a few iterations and the class of largest entry can be considered as the identification result. Using the concept of sparse level which we have defined before, the iteration processing can be stopped when a preset sparse level threshold reaches after some iteration times.

Proof. It can be proved that for a system of linear equations $Ax = y$, if a solution x exists obeying

$$\|x\|_0 \leq \frac{1}{2} \left(1 + \frac{1}{\mu(A)} \right). \quad (8)$$

OMP run with threshold parameter $\varepsilon_0 = 0$, is guaranteed to find it exactly. This theorem is only valid when test sample y can be represented by the linear combination of A exactly,

$$y = Ax = \sum x_t a_t. \quad (9)$$

Suppose that, $\mu(A)$ is equal or greater than some value close to 1, and after some iterations, the sparse level of the solution is still greater than p , and we only reserve the $(1 + 1/\mu(A))/2$ greater entries; hence

$$y = \left\{ 0, 0, \dots, x_i, 0, \dots, \frac{x_i}{p}, \dots \right\}. \quad (10)$$

It can be proved that if a sparse vector satisfies the sparsity constraint $\|x\|_0 < (1 + 1/\mu(A))/2$ and gives a representation of y to within error tolerance ε , every solution x must obey

$$\|x^\varepsilon - x\|_2^2 \leq \frac{4\varepsilon^2}{1 - \mu(A)(2\|x\|_0 - 1)}. \quad (11)$$

Note that as we reconstruct y as one image in A , the smallest error between y and the recovery image is $\mu(A)$, so ε is larger than $1 - \mu(A)$. Therefore,

$$\begin{aligned} \|x^\varepsilon - x\|_2^2 &\leq \frac{4(1 - \mu(A))^2}{1 - \mu(A)(1 + (1/\mu(A)) - 1)} \\ &= 4[1 - \mu(A)]^2 < 4\varepsilon^2. \end{aligned} \quad (12)$$

□

4. Processing for Images under Random Corruption and Contiguous Occlusion

To almost all sparse representation algorithms, as based on errors of pixel in the corresponding position, it is brittle for them to cope with samples under large level of occlusion or corruption [7, 16].

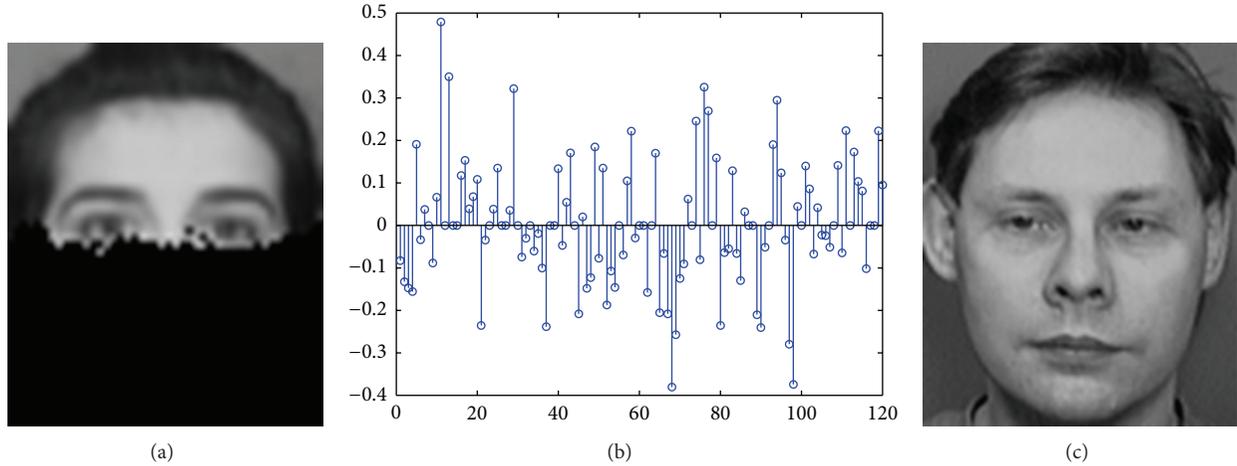


FIGURE 1: (a) is an input sample with contiguous occlusion. (b) is the sparse solution based on the OMP algorithm. (c) is the reconstructed image, whose class is the same as the class of the largest entry of the sparse solution. Obviously, the solution is not sparse enough and we have got a wrong answer due to the occluded part in y .

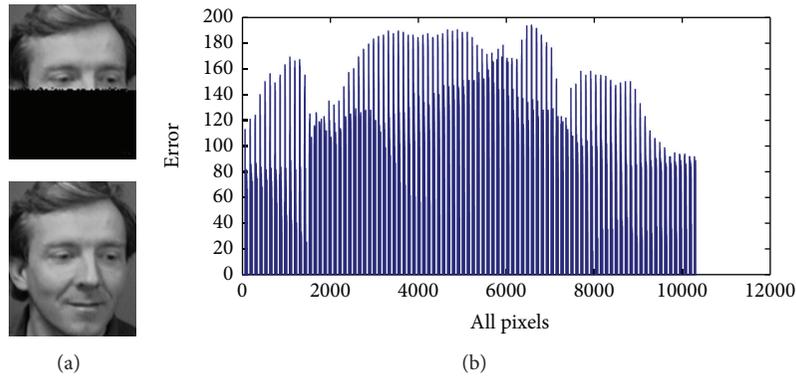


FIGURE 2: Error between an occluded image and a “clean” one. We reshape the pixels in the two samples (a) into vectors and calculate the difference (b). Obviously, the distribution of error is like a comb, which indicates that the error exists in a few pixels and the others are “clean.”

4.1. Necessity of Preprocessing of Test Samples and Training Dictionary. When the test sample y is occluded seriously, the error between y and each column of A increases, which means each element of vector e rises, where $e_i = y - a_i$. Hence, both the largest coefficient and the sparse level after some fix iteration times can hardly reach their respective predetermined thresholds. On condition that we employ A and y directly, as shown in Figure 1, the solution we get can be dense, which will surely weaken the superiority of sparse representation. Therefore it is necessary to process input sample y and training dictionary A firstly.

It has been proposed that this kind of issue can be solved by block partitioning, which means to partition the image into blocks and process each block independently. The results for individual blocks are then aggregated. This method is only valid to images under contiguous occlusion. And the processing takes quite a long time since it transforms the classification to several subproblems. Let us think about how human brains handle the face images disguised by scarf or glasses. We make out the scarf or glasses in the image and then neglect these parts which are unrelated to face. And

our judgment of the person in the image depends on the other parts which we regard as face. Imitating the method human brain dealing with this kind of image, we can propose to preprocess the input image and training matrix before applying them to the OMP algorithm. No matter the random corruption or the contiguous occlusion part can be regarded as noise in the image, we can uniformly reject this part and only pay attention to the other parts.

4.2. How to Identify a Corrupted or Occluded Test Sample Automatically. Firstly, the system should automatically identify whether the test image is the one with partial occlusion or corruption. The errors between the occlusion and corruption image which belong to a class in the training dictionary or an invalid one are both quite large. But the error of an occluded one has some distinguishing features—the error between pixels focuses on only a few pixels; the others’ error is quite small. So if the error vector contained some elements which are closed to 0 and the variance of the error vector is

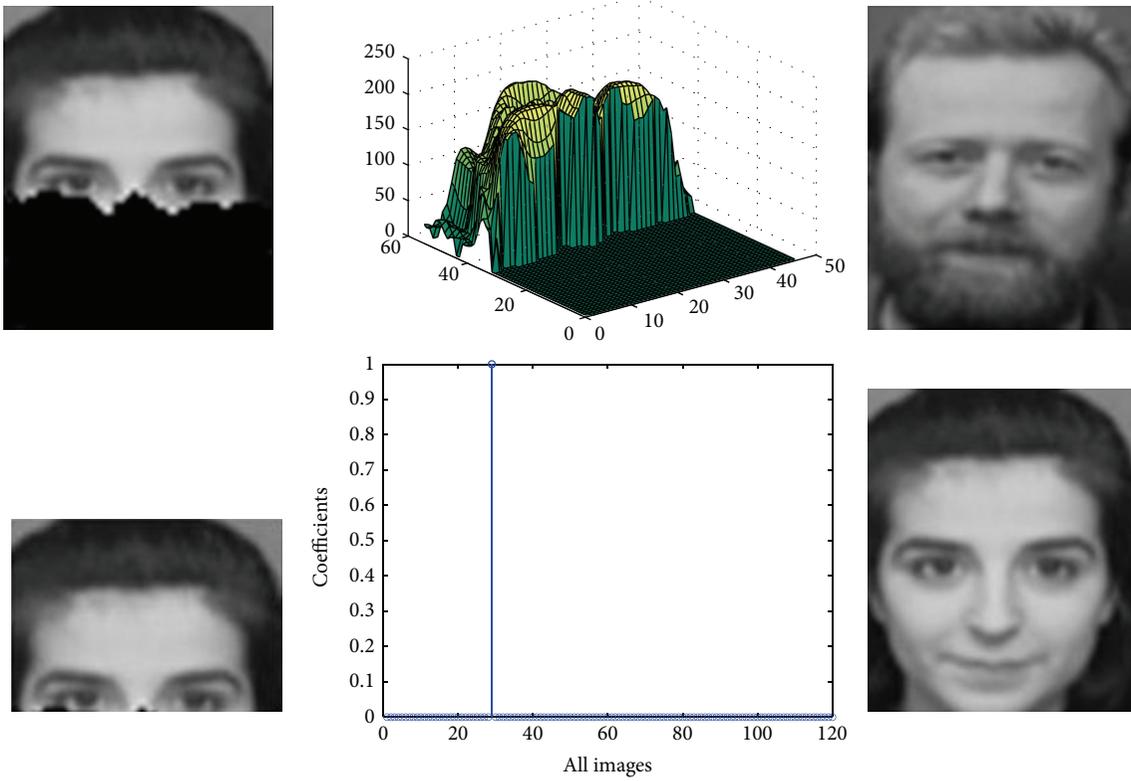


FIGURE 3: Applying the occluded image (left top) directly to the algorithm, we get the wrong answer because part of the noise is too large for the system. By rejecting some pixels in the test image and shrinking the training matrix, respectively, the right person can be recognized.

large enough, $\sigma_e > \sigma_0$, we can regard the test sample as a partial occlusion or corruption one, as shown in Figure 2.

4.3. Images Preprocessing to Corrupted or Occluded Samples. Then let us discuss how to use extract the “clean” pixels and remove the invalid ones. To begin with, we can define an error threshold ξ between a particular pixel in test sample y and respective pixels of elements in A . If the minimal error between the pixel in y and respective pixel in one class of A is larger than ξ ,

$$\min(y_i - a_{ij}) > \xi. \quad (13)$$

We can regard this pixel as an invalid one and remove it from y as well as the respective one from A . Then we get the new training matrix A_1 and test vector y_1 , whose “noise pixels” have been filtered. So the problem has been transformed into finding a sparse solution x_1 subject to

$$y_1 = A_1 \cdot x_1. \quad (14)$$

And the improved OMP algorithm can just be used in this new generated equation. One example is given in Figure 3. This will surely enhance the recognition rate of the system. Since this method does not involve any constraint conditions regarding noise distribution, the issue of random corruption and contiguous occlusion can be solved simultaneously. “Shrinking” of A and y also makes the computing speed faster.

5. Experimental Results

In this section, we apply our algorithm on ORL database for face recognition. We first test the recognition rate and elapsed time of the algorithm, compared to the state-of-art algorithm to find the sparse solution. We then examine the identifying performance to corruption and occlusion. Finally, we simulate the real situation and check up the robustness under various disguises.

5.1. Performance on ORL Database and Yale Database. ORL database consists of 400 frontal images for 40 individuals. Samples in this database include facial variation like various expressions and postures, which can be obstacles or challenges for the system to find the true class of the test images.

We test the face recognition rate and elapsed time of the algorithm by 10-fold cross-validation. In other words, for each test, the training dictionary is consisted of 40 classes of 360 images (9 samples per class) and the remaining 40 images are test samples. All these images have been simply downsampled, without any particular feature extraction. We compared the result with original OMP, together with some state-of-art algorithms aiming at sparse solution—primal augmented Lagrangian method (PALM), dual augmented lagrangian method (DALM), fast iterative soft-thresholding algorithm (FISTA), and truncated Newton interior-point Method (TNIPM). All results of each test and the average are in Table 1.

TABLE 1: Comparison on ORL database.

Test number	Elapsed time per test image(s)					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	1.0957	2.6292	0.3548	0.9861	33.1391	0.0278
2	1.3422	2.6434	0.3628	1.0106	35.4203	0.0248
3	1.5197	2.6408	0.3607	0.9211	34.7244	0.0233
4	1.5638	2.6922	0.3661	0.9212	35.2557	0.0253
5	1.2596	2.6749	0.3655	0.8308	36.5687	0.0238
6	1.4026	2.6485	0.3635	0.8343	36.8514	0.0259
7	1.3802	2.7209	0.3682	0.8390	36.7581	0.0218
8	1.2477	2.7298	0.3742	1.0580	36.8528	0.0208
9	1.4948	2.7602	0.3635	0.8528	35.9073	0.0213
10	1.3220	2.6478	0.3670	1.04	38.6038	0.0214
Average(s)	1.3628	2.6788	0.3646	0.9294	36.008	0.0236
Test number	Recognition rate					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	62.5%	87.5%	90%	87.5%	77.5%	95%
2	70%	90%	90%	90%	82.5%	92.5%
3	77.5%	92.5%	92.5%	92.5%	80%	95%
4	80%	90%	92.5%	90%	80%	95%
5	70%	92.5%	92.5%	92.5%	85%	97.5%
6	77.5%	92.5%	92.5%	92.5%	82.5%	95%
7	75%	92.5%	95%	95%	87.5%	95%
8	67.5%	92.5%	92.5%	92.5%	85%	97.5%
9	82.5%	95%	95%	95%	85%	97.5%
10	75%	92.5%	95%	92.5%	82.5%	95%
Average	73.75%	91.75%	92.75%	92%	82.75%	95.5%

TABLE 2: Comparison on Yale database.

Test number	Elapsed time per test image(s)					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	1.9440	1.3520	5.9824	20.1194	12.433	0.7253
2	1.6345	1.3967	6.4434	17.1639	11.1681	0.7690
3	1.8704	6.3440	6.3742	17.4002	11.3312	0.7831
4	1.9500	2.0409	7.0701	18.1936	11.0440	0.7920
5	1.6420	1.3959	6.1629	16.8426	10.5832	0.7215
6	1.3926	1.3948	6.1054	15.7372	11.6463	0.7824
7	1.5375	1.3638	7.3222	15.9961	13.0719	0.6903
8	1.3519	1.4371	7.8755	15.6780	11.1364	0.8467
Average(s)	1.6654	2.0907	6.6670	17.1414	11.5518	0.6731
Test number	Recognition rate					
	PALM	DALM	FISTA	TNIPM	OMP	Improved OMP
1	52.96%	47.37%	92.11%	55.92%	78.95%	92.43%
2	53.29%	50.33%	95.39%	90.13%	81.58%	92.76%
3	51.97%	52.63%	92.76%	87.17%	83.22%	90.46%
4	53.29%	51.64%	92.43%	91.12%	85.53%	95.72%
5	47.37%	49.34%	94.41%	91.78%	79.93%	94.74 %
6	50%	51.64%	96.38%	93.75%	82.89%	96.71%
7	49.34%	47.37%	93.42%	92.76%	82.89%	92.43%
8	51.64%	50%	95.39%	94.74%	86.51%	95.72%
Average	51.23%	50.04%	94.04%	87.17%	82.69%	93.87%

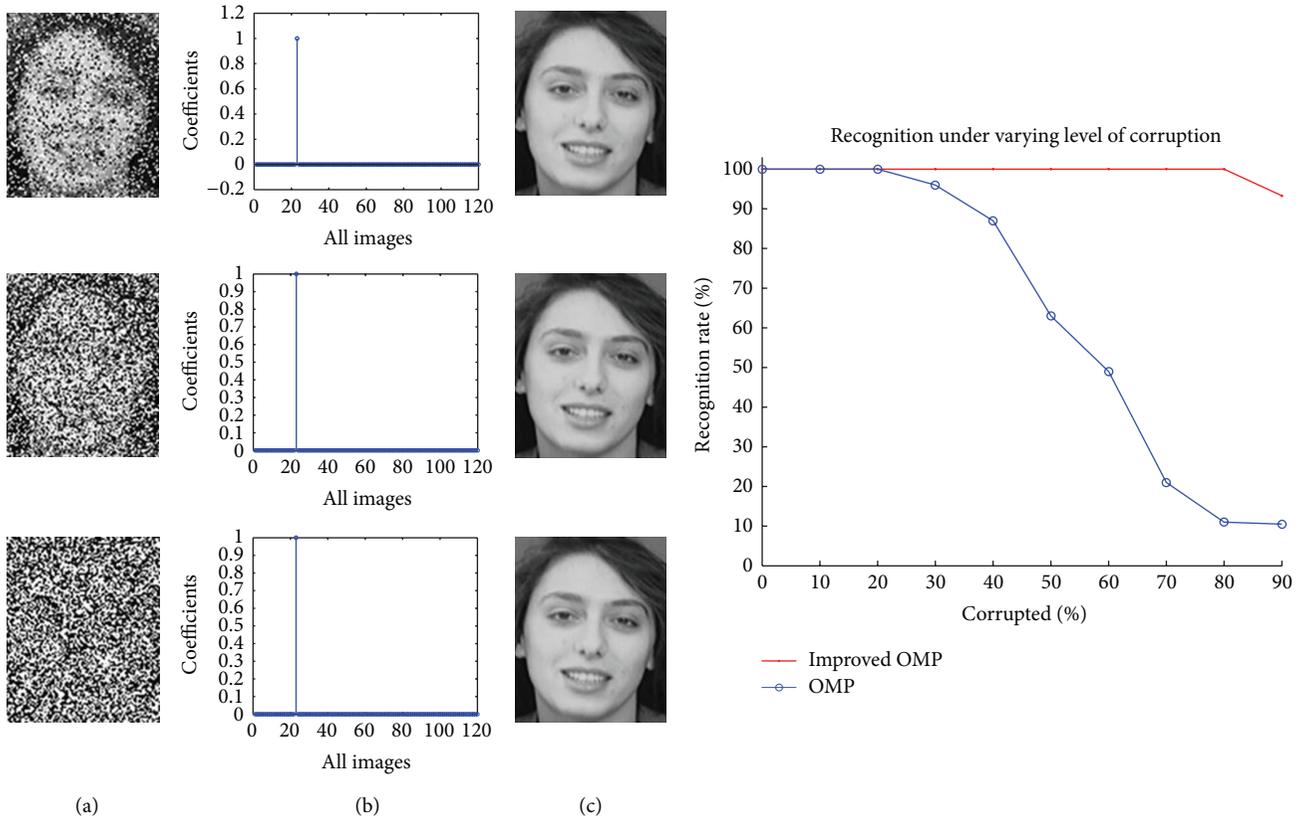


FIGURE 4: Recognition under random corruption. Left: (a) test images y from ORL database, with random corruption. Top row: 30% of pixels are corrupted, middle row: 60% corrupted, bottom row: 90% corrupted. (b) Estimated sparse coefficients. (c) Reconstructed images. We can see that our algorithm outperforms the original OMP.

TABLE 3: Comparison on FERET database.

Test number	Elapsed time per test image(s)					Improved OMP
	PALM	DALM	FISTA	TNIPM	OMP	
1	13.6277	0.6245	3.7790	8.8048	10.1515	0.3854
2	13.6254	0.6191	3.8656	9.4325	10.4445	0.3325
3	18.9904	0.6191	3.8538	8.6242	10.2513	0.2858
4	15.2641	0.6316	4.0842	9.0337	10.2262	0.3229
5	15.2024	0.6364	4.0117	9.0880	10.4940	0.4109
6	13.1985	0.6125	3.8448	8.7168	10.2518	0.4428
7	14.5053	0.6365	3.7359	8.8999	10.8678	0.2599
Average(s)	14.9163	0.6257	3.8821	8.9428	10.3839	0.3486
Test number	Recognition rate					Improved OMP
	PALM	DALM	FISTA	TNIPM	OMP	
1	40%	90.67%	92.67%	90.67%	51.33%	93.33%
2	48.67%	88%	92.67%	88%	50%	92%
3	45.33%	65.33%	68.67%	65.33%	77.33%	92.67%
4	38.67%	84.67%	88%	84.67%	72.67%	90.67%
5	44.67%	86.67%	92%	86.67%	74.67%	93.33%
6	54.67%	80%	86%	80%	57.33%	89.33%
7	42.67%	90%	93.33%	90%	64%	94.67%
Average	44.95%	83.62%	87.62%	83.62%	63.90%	92.29%

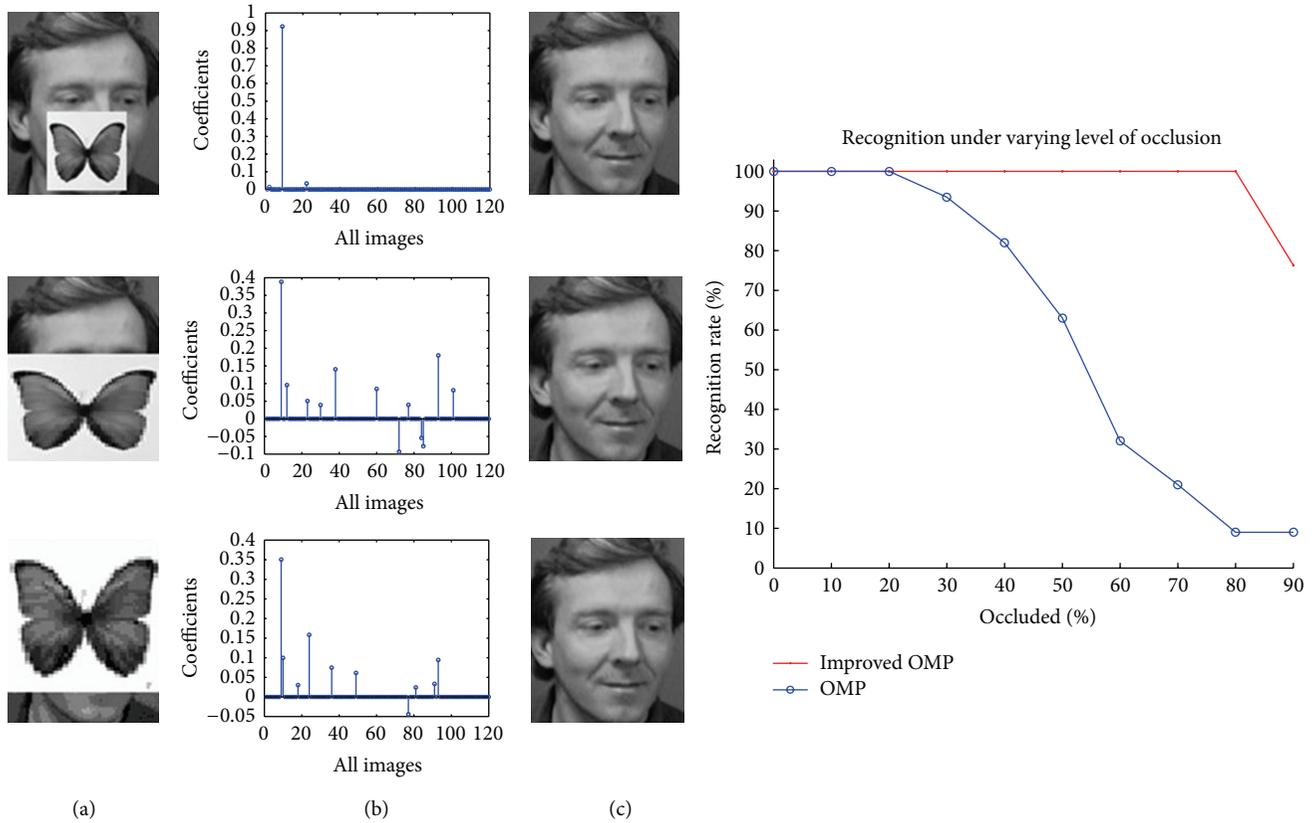


FIGURE 5: Recognition under varying levels of contiguous occlusion. (a) Top row: 30% occluded test sample, middle row: 60% occluded, and bottom row: 90% occluded. (b) Estimated sparse coefficients. (c) Reconstructed images.

As shown in Table 1, the recognition rate of improved OMP is the highest. Meanwhile, elapsed time per sample outperforms others.

We also compared the algorithm with others in the Yale database. Yale database contains 2432 frontal images for 38 individuals, which were captured under various laboratory-controlled lighting conditions. 8-fold cross-validation has been taken in this database, with 56 images of each class as training samples, and the other 304 images as test samples. Just as the tests on ORL database, the images have only been downsampled to construct the training dictionary to make the problem $Ax = y$ become undetermined. The results of the 8 tests and average values are in Table 2.

In Table 2, although the recognition rate of FISTA is higher than our algorithm, its run time is almost tenfold compared with the improved OMP.

Similar experiments were performed on FERET database. Compared with the other face databases we mentioned above, this database includes more variations like postures and facial expressions. We ran 7-fold cross-validation, with 150 classes (6 samples per class) in this database as training database, the other 150 samples as test samples in each test. Data in Table 3 reflects the comparison of these algorithms.

In Table 3, we can see that the average recognition rate of improved OMP is the highest in tests of FERET database and its run time is much shorter.

We get these data in MATLAB on a typical 2.40 GHz PC with quad-core processor. To be fair, both the training dictionary and the test samples are the same to all algorithms. And all identified results depend on the class of the largest coefficient in the sparse solution. The improved OMP algorithm greatly reduces the run time, and its recognition rates are relatively well.

5.2. Recognition despite Occlusion and Corruption

5.2.1. Recognition under Random Corruption.

We first test the robust version of our algorithm for samples under random corruption. We add salt and pepper noise of different intensities to samples in the database to generate the test samples. Figure 4 plots the recognition results of the robust version of the OMP and applying OMP directly to the test samples.

From Figure 4, when the face imaged is 90% corrupted by the noise, although we can hardly identify it as a face image, the algorithm still reconstructed the right image. The right line graph in Figure 4 indicates that this method performs quite well under the condition of large percent of random corruption.

5.2.2. Recognition under Continuous Occlusion.

We add an irrelevant image with different sizes to the samples in the

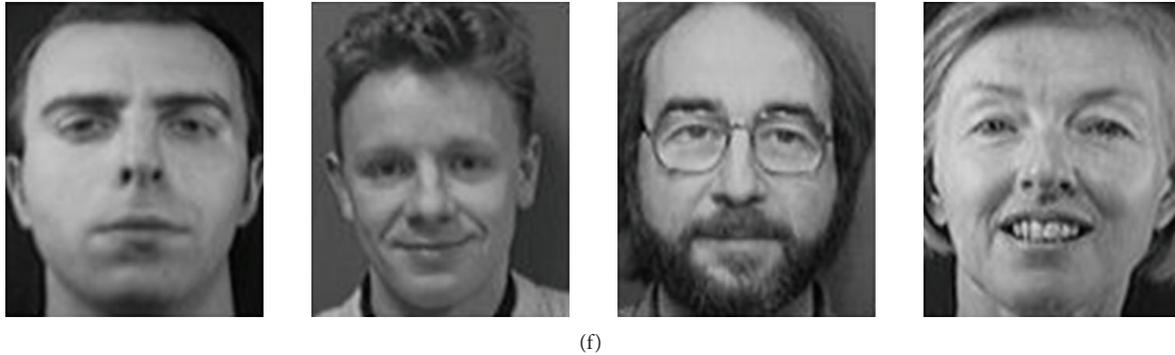


FIGURE 6: Recognition despite disguise like sunglasses, scarf, or respirator. (a) Test images with sunglasses or scarf. (b) Estimated sparse coefficients. (c) Reconstructed images. (d) Test images with sunglasses and scarf. (e) Estimated sparse coefficients. (f) Reconstructed images.

database and treat them as test image to test the robustness under continuous occlusion. Figure 5 indicates the performance.

We can see in the right of Figure 5 that the improved OMP significantly outperforms the original one, which shows its robustness of occlusion.

5.3. Recognition despite Disguise. Face photos taken in a real world scenario often contained glasses and scarf, which makes it harder for the system to identify the right person. Now let us examine the performance of the algorithm under these kinds of situation. Our test images are also from ORL database and we add glasses and scarf pictures to the samples.

Figure 6 shows that the algorithm also has a quite well performance on the real situation—disguised test samples. We constructed a disguised test sample database based on ORL database of 40 samples with sunglasses and scarf. And the recognition rate reached 95%.

6. Conclusions

In this paper, we proposed a fast and robust algorithm based on OMP algorithm. We first discussed the disadvantage of OMP algorithm to solve the face recognition problem. Then an improved method is proposed to make the elapsed time become much shorter to identify a test image. We also tried to enhance robustness to the occluded and corrupted test samples by extracting the “noiseless” pixel and reduce the elements in both the test image and training dictionary, respectively. Finally, we prove this method by experiments on ORL database, Yale database, and FERET database.

One further work is to enhance the robustness in situation under various kinds of misalignment and postures. We may further reduce the constrained condition and apply this method to object recognition.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (no. 61379010) and Natural Science Basic Research Plan in Shaanxi Province of China (no. 2012JQ1012).

References

- [1] J. Wright, A. Yang, A. Ganesh et al., “Robust face recognition via sparse representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
- [2] Y. Su, S. G. Shan, X. L. Chen et al., “Hierarchical ensemble of global and local classifier for face recognition,” *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 273–292, 2009.
- [3] A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Fast l_1 -minimization algorithms and an application in robust face recognition: a review,” in *Proceedings of the International Conference on Image Processing*, 2010.
- [4] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, “Sparse representation for computer vision and pattern recognition,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.
- [5] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [6] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [7] A. Martinez, “Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 748–763, 2002.
- [8] D. Donolo and M. Elad, “Optimal sparse representation in general(nonorthogonal) dictionaries via minimization,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [9] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal l_1 -norm solution is also the sparsest solution,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

- [10] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.
- [11] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
- [12] A. Yang, M. Gastpar, R. Bajcsy, and S. Sastry, "Distributed sensor perception via sparse representation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1077–1088, 2010.
- [13] D. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.
- [14] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [15] F. Sanja, D. Skocaj, and A. Leonardis, "Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, 2006.
- [16] J. Wright and Y. Ma, "Dense error correction via ℓ^1 -minimization," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3540–3560, 2010.

Research Article

Robust Calibration of Cameras with Telephoto Lens Using Regularized Least Squares

Mingpei Liang,¹ Xinyu Huang,¹ Chung-Hao Chen,² Gaolin Zheng,¹ and Alade Tokuta¹

¹North Carolina Central University, Durham, NC 27707, USA

²Old Dominion University, Norfolk, VA 23529, USA

Correspondence should be addressed to Xinyu Huang; xinyu.huang@ncu.edu and Chung-Hao Chen; cxchen@odu.edu

Received 3 October 2013; Accepted 24 December 2013; Published 11 February 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Mingpei Liang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cameras with telephoto lens are usually used to recover details of an object that is either small or located far away from the cameras. However, the calibration of this kind of cameras is not as accurate as the one of cameras with short focal lengths that are commonly used in many vision applications. This paper has two contributions. First, we present a first-order error analysis that shows the relation between focal length and estimation uncertainties of camera parameters. To our knowledge, this error analysis with respect to focal length has not been studied in the area of camera calibration. Second, we propose a robust algorithm to calibrate the camera with a long focal length without using additional devices. By adding a regularization term, our algorithm makes the estimation of the image of the absolute conic well posed. As a consequence, the covariance of camera parameters can be reduced greatly. We further used simulations and real data to verify our proposed algorithm and obtained very stable results.

1. Introduction

In various vision based applications, a camera with a telephoto lens is often useful to acquire detailed information of objects. It could capture high resolution face images for the purpose of recognition and reconstruction even when a user is at a distance [1]. It also could obtain eye images with rich iris textures when a user is several meters away from the camera [2, 3]. In [4], a telephoto lens is used to observe objects under the influence of optical turbulence. By combining with a wide-angle camera, a robotic vision system has been shown in [5], which is suitable for remote surveillance or minimally invasive surgical interventions that could have a higher resolution than typical commercial endoscopes. As the field of view of a telephoto lens could only have a few degrees (e.g., around 8 degrees for a 300 mm telephoto lens), in order to either track objects or reconstruct complete views, an accurate estimation of camera parameters is required.

In the photogrammetry community, camera calibration usually is done by computing the projection matrix using accurate 3D points and corresponding 2D observations [6, 7]. However, in practice, it could be difficult or expensive to build an object with accurate coordinates, especially in a large

working space. In the area of computer vision, the calibration technique [8] that requires only a planar pattern (e.g., a checkerboard pattern) is widely used. In this technique, a planar pattern is placed with different orientations and at different distances from the camera. Homographies are estimated between the planar pattern and its observations. These homographies could form a homogeneous system that is used to solve the image of the absolute conic. The intrinsic and extrinsic parameters are then computed by using the estimated homographies and the image of the absolute conic. In the final step, the maximum likelihood estimation (MLE) is applied to estimate the radial distortion and refine the intrinsic and extrinsic parameters by minimizing geometric errors. This technique has further been evaluated with respect to image noise level, number of planes, and orientation of the model plane. Various autocalibration techniques [9, 10] are also proposed to estimate fixed or varying intrinsic parameters without predefined calibration patterns. The basic idea is that the absolute conic is fixed when a camera is moving rigidly.

Since focal lengths of cameras in many vision applications are relatively short, most of existing algorithms consider image noise as a major source of estimation uncertainties

and limited research has been conducted on the uncertainties caused by focal length [11, 12]. In [11], Strobl et al. found that the narrow field of view makes the calibration more difficult due to lack of required evidence on perspective. In order to improve the calibration accuracy, the camera with a narrow field of view is mounted on a robotic manipulator from which the rigid motions can be read. These rigid motions could provide more constraints for solving the intrinsic parameters and a relative geometric relation between the camera and the robotic manipulator. Similarly, a pan-tilt unit could be used during the calibration as shown in [12].

There are mainly two contributions in this paper. Firstly, we present a first-order error analysis that shows the relation between estimation uncertainties and focal length. Although authors in [11, 12] briefly described the calibration problem caused by long focal length, the error analysis with respect to focal length has not been studied so far. Secondly, we propose a robust algorithm without using additional devices, which is based on the regularization term defined by the prior of the image of the absolute conic.

The remaining of this paper is organized as follows. Section 2 introduces necessary notations and background of existing algorithm using estimated homographies. Section 3 gives the error propagation from image noise to camera parameters. Our calibration algorithm is proposed in Section 4. Section 5 shows the experiments on simulation and real data. The conclusion is given in Section 6.

2. Notation and Background

In this section, we start with the notation and then briefly introduce the calibration technique proposed in [8].

The homography between the planar pattern and image plane is denoted by $H = [h_1 h_2 h_3]$, the intrinsic matrix is given by

$$K = \begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where (u_0, v_0) is the coordinates of the principal point, $\alpha_x = m_x f$ and $\alpha_y = m_y f$ are scale factors, and m_x and m_y are the number of pixels per unit distance in image along x and y directions. The image of the absolute conic is $\omega = K^{-T} K^{-1}$.

Given an image of a planar pattern, two constraints can be imposed on the intrinsic parameters, $h_1^T \omega h_2 = 0$ and $h_1^T \omega h_1 = h_2^T \omega h_2$. Therefore, a constrained optimization can be formed by

$$\begin{aligned} \min_{\omega_v} \quad & \|A\omega_v\|, \\ \text{s.t.} \quad & \|\omega_v\| = 1, \end{aligned} \quad (2)$$

where ω_v is a 6×1 vector extracted from ω and A is a $2n \times 6$ matrix constructed from entries of H . The intrinsic matrix K is computed by the Cholesky factorization. Since the close form solution is obtained through minimizing algebraic errors, the maximum likelihood is further applied to refine the results by minimizing geometric errors.

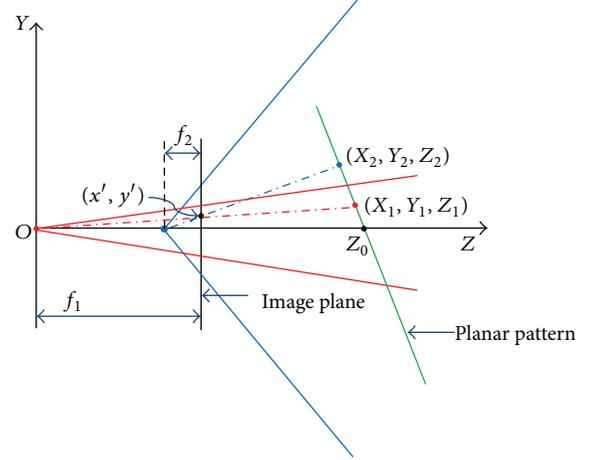


FIGURE 1: Two cameras with the same image plane and different focal lengths.

The calibration performance with respect to image noise level, the number of planes, and the orientation of the model plane are also evaluated in [8]. Based on the computer simulations, the errors increase linearly with the image noise level and decrease when more images are used. The best orientation of the model plane is around 45 degrees.

3. Covariance of the Estimated Intrinsic Parameters

In order to find out the relation between focal length and uncertainties of camera parameters, it is not sufficient to only have a point estimate of the parameters. In this section, we present a first-order approximation to compute covariance of estimated parameters.

Let us consider two cameras with different focal lengths f_1 and f_2 (assume $f_1 > f_2$) and sharing a same image plane. The origin O is located at the center of projection of camera 1. Through a single point (x', y') on the sensor, camera 1 observes a 3D point $\mathbf{X}_1 = (X_1, Y_1, Z_1)^T$ and camera 2 observes a 3D point $\mathbf{X}_2 = (X_2, Y_2, Z_2)^T$, while \mathbf{X}_1 and \mathbf{X}_2 are located on a same planar pattern. We also define a transformation M that transforms the 3D points \mathbf{X}_1 and \mathbf{X}_2 to a coordinate system on the planar pattern such that new depths \tilde{Z}_1 and \tilde{Z}_2 are equal to 0. Let us denote $\tilde{\mathbf{x}}_1 = M \cdot (X_1, Y_1)^T$ and $\tilde{\mathbf{x}}_2 = M \cdot (X_2, Y_2)^T$. The configuration is shown in Figure 1.

As this configuration consists of the same image plane, same 2D observations, and same orientations and locations of the planar pattern, the major difference between two cameras is the focal lengths. For simplicity, we assume all the pixels are square, we have

$$(x', y')^T \mapsto \frac{f_1}{Z_1} (X_1, Y_1)^T. \quad (3)$$

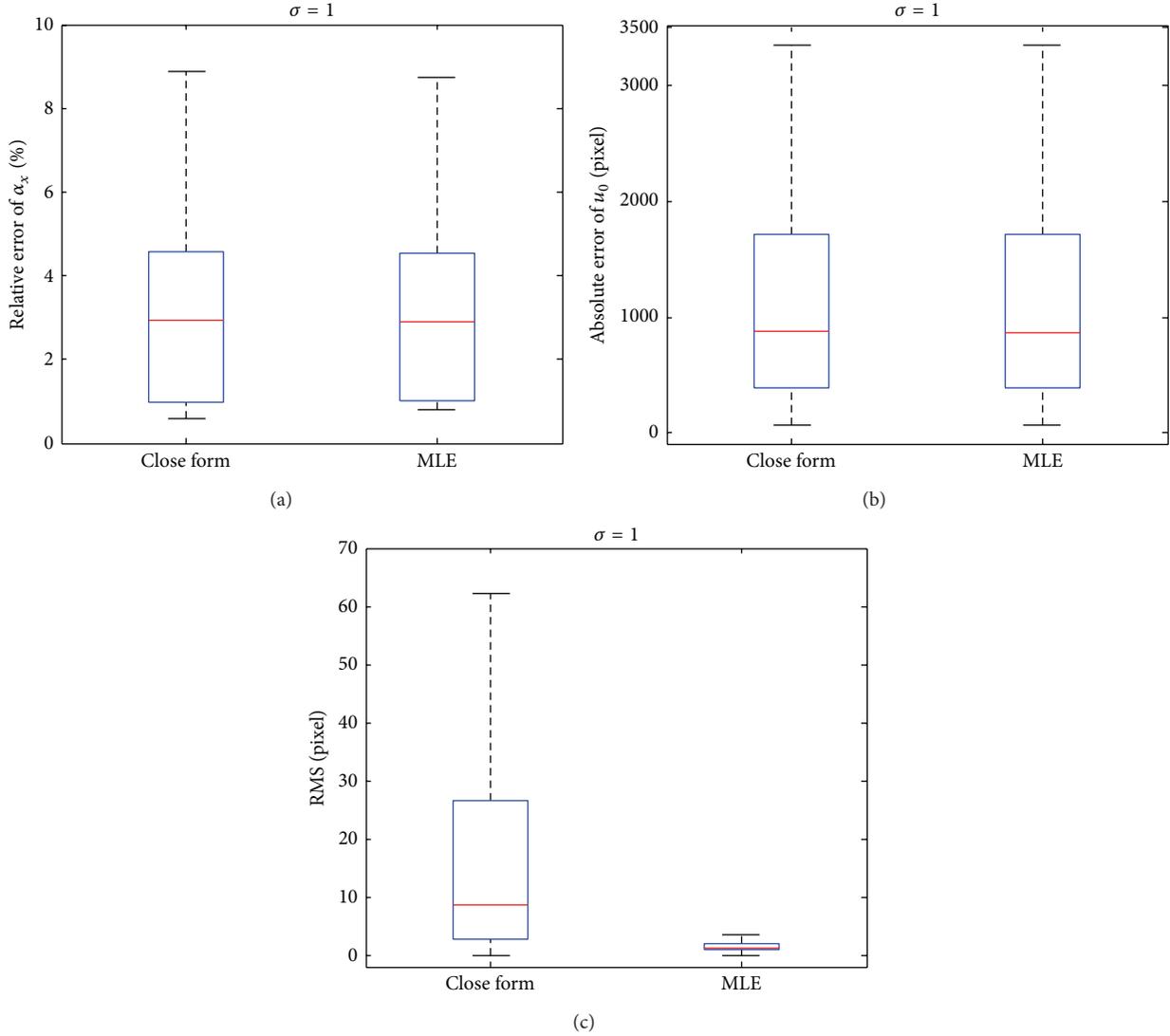


FIGURE 2: Comparison of uncertainties between the close form solution and the MLE solution [8] ($f = 300$ mm, $\alpha_x = 26034$, and $\alpha_y = 29163$). The image noise is $\mu = 0$ with $\sigma = 1$. The mean relative errors of α_x are 4.74% and 4.72%, the mean absolute errors of u_0 are 1717 and 1689 pixels, and the RMS reprojection errors are 19.5 and 3.2 pixels, for the close form and the MLE solutions, respectively.

Since the center of projection of camera 2 is at $(f_1 - f_2, 0)^T$, through the same point (x', y') , we have

$$(X_2, Y_2)^T \mapsto \frac{(Z_2 + f_2 - f_1)}{f_2} (x', y')^T. \quad (4)$$

Putting this together with (3) leads to the formula

$$\bar{\mathbf{x}}_2 = M \cdot (X_2, Y_2)^T \mapsto \frac{f_1}{f_2} \cdot \frac{Z_2 + f_2 - f_1}{Z_1} \cdot \bar{\mathbf{x}}_1. \quad (5)$$

Let $\mathbf{x} = (x, y)$ be the image coordinate of the point (x', y') on the sensor. Assuming that two cameras have same image resolution and the principal point, \mathbf{x} is same for both cameras. Let us further assume that the noise is limited to

the observed image with covariance $\Sigma_{\mathbf{x}}$, the covariance of the intrinsic parameters $\Sigma_{\mathbf{k}}$ is

$$\Sigma_{\mathbf{k}} = J_{\mathbf{k}} J_{\omega_v} (J_{\mathbf{h}}^T \Sigma_{\mathbf{x}}^{-1} J_{\mathbf{h}})^+ J_{\omega_v}^T J_{\mathbf{k}}^T, \quad (6)$$

where $J_{\mathbf{k}}$, J_{ω_v} , and $J_{\mathbf{h}}$ are the Jacobian matrices evaluated at $\hat{\mathbf{k}}$, $\hat{\omega}_v$, and $\hat{\mathbf{h}}$, respectively. \mathbf{k} , ω_v , and \mathbf{h} are the vectors made up of the entries of the intrinsic matrix K , the image of the absolute conic ω , and 2D homography H . As $\Sigma_{\mathbf{x}}$, $J_{\mathbf{k}}$, and J_{ω_v} only depend on \mathbf{k} and ω_v and are same for both cameras, we only need to analyze the relation between $J_{\mathbf{h}}$ of two cameras. The Jacobian matrix $J_{\mathbf{h}}^i = \partial \mathbf{x}_i / \partial (h)$ for i th observed point is given by

$$J_{\mathbf{h}}^i = \frac{\partial \mathbf{x}_i}{\partial \mathbf{h}} = \frac{1}{w'_i} \begin{bmatrix} \bar{\mathbf{x}}_i^T & \mathbf{0}^T & -x_i \bar{\mathbf{x}}_i^T \\ \mathbf{0}^T & \bar{\mathbf{x}}_i^T & -y_i \bar{\mathbf{x}}_i^T \end{bmatrix}. \quad (7)$$

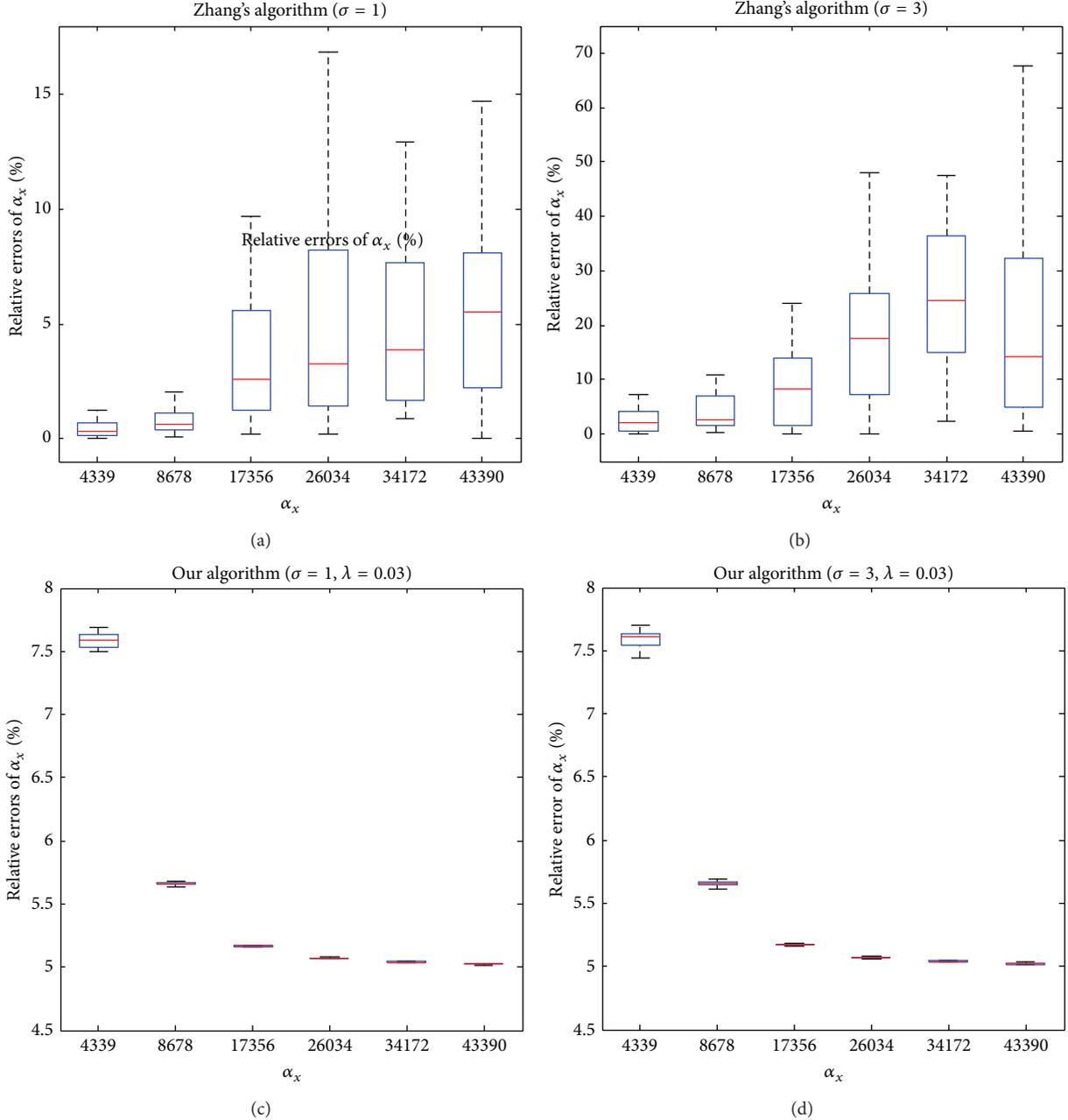


FIGURE 3: Comparison of uncertainties of α_x between our algorithm and [8].

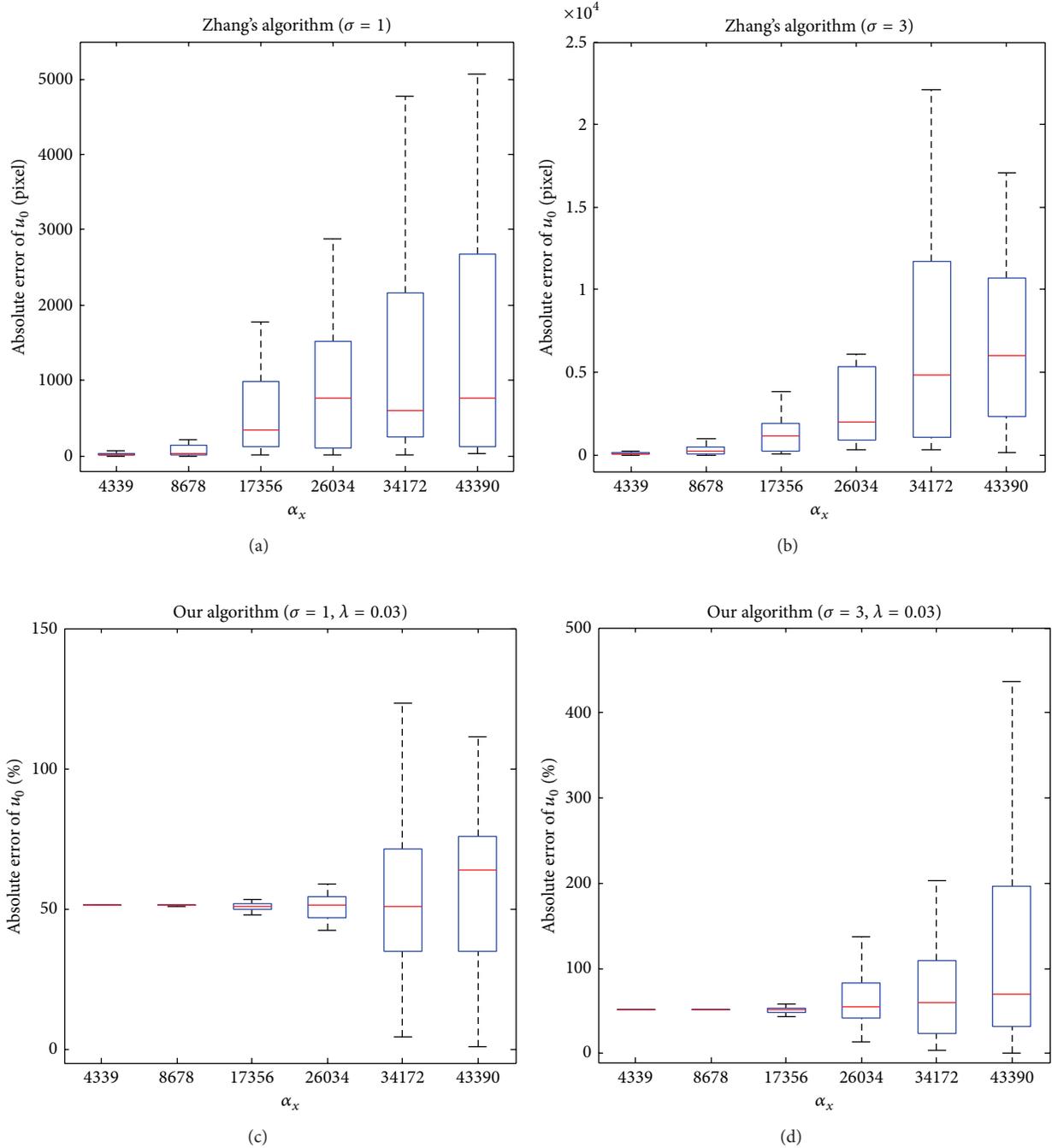
Based on (5) and (7), it is not difficult to find that

$$\begin{aligned} \Sigma_{\mathbf{h}_2} &= \left(\sum_i J_{\mathbf{h}_2, i}^T \Sigma_i^{-1} J_{\mathbf{h}_2, i} \right)^+ \\ &= \left(\frac{f_2}{f_1} \right)^2 \left(\sum_i \left(\frac{Z_{2,i} + f_2 - f_1}{Z_{1,i}} \right)^2 \cdot J_{\mathbf{h}_1, i}^T \Sigma_i^{-1} J_{\mathbf{h}_1, i} \right)^+, \end{aligned} \quad (8)$$

where \mathbf{h}_1 and \mathbf{h}_2 are vectors made up of the entries of homographies between the planar pattern and camera 1 and 2, $Z_{1,i}$ and $Z_{2,i}$ are depths of i th 3D points observed by cameras

1 and 2, respectively. Σ_i is the covariance matrix of the i th measured image point. From this equation, we can find that the covariance of 2D homography is also affected by focal length and orientation and depth of the planar pattern.

Notice that $f_2 - f_1$ is usually far less than depths $Z_{1,i}$ and $Z_{2,i}$; we could approximate $(Z_{2,i} + f_2 - f_1)/Z_{1,i}$ by $Z_{2,i}/Z_{1,i}$. Since image resolution and focal length are fixed for two cameras, in order to reduce the uncertainties of the estimated homography, one possible direction is to increase the pan and tilt angles of the planar pattern so that $Z_{2,i}/Z_{1,i}$ is very close to 0. However, as mentioned in [8], the best orientation is around 45 degrees, which means that this ratio cannot be


 FIGURE 4: Comparison of uncertainties of u_0 between our algorithm and [8].

very small. This direction is also not feasible in practice due to the limited depth of field. When the region of the planar pattern is outside the depth of field, the sharpness of the region decreases and image noise modeled by Σ_x increases. Moreover, as the planar pattern could be considered as being uniformly distributed within the field of view of a camera, expectations of both $Z_{1,i}$ and $Z_{2,i}$ are close to the depth Z_0 shown in Figure 1. Therefore, for simplicity, it is reasonable to approximate $Z_{2,i}/Z_{1,i}$ as 1. As a result, (8) could be simplified

to $\Sigma_{h_2} = (f_2/f_1)^2 \cdot \Sigma_{h_1}$, and the relation between covariance matrices of intrinsic matrices of two cameras is given by

$$\Sigma_{k_1} = \left(\frac{f_1}{f_2}\right)^2 \cdot \Sigma_{k_2}. \quad (9)$$

Therefore, uncertainties of intrinsic parameters \mathbf{k} increase when focal length increases. Since extrinsic parameters for each image can be determined by intrinsic parameters and

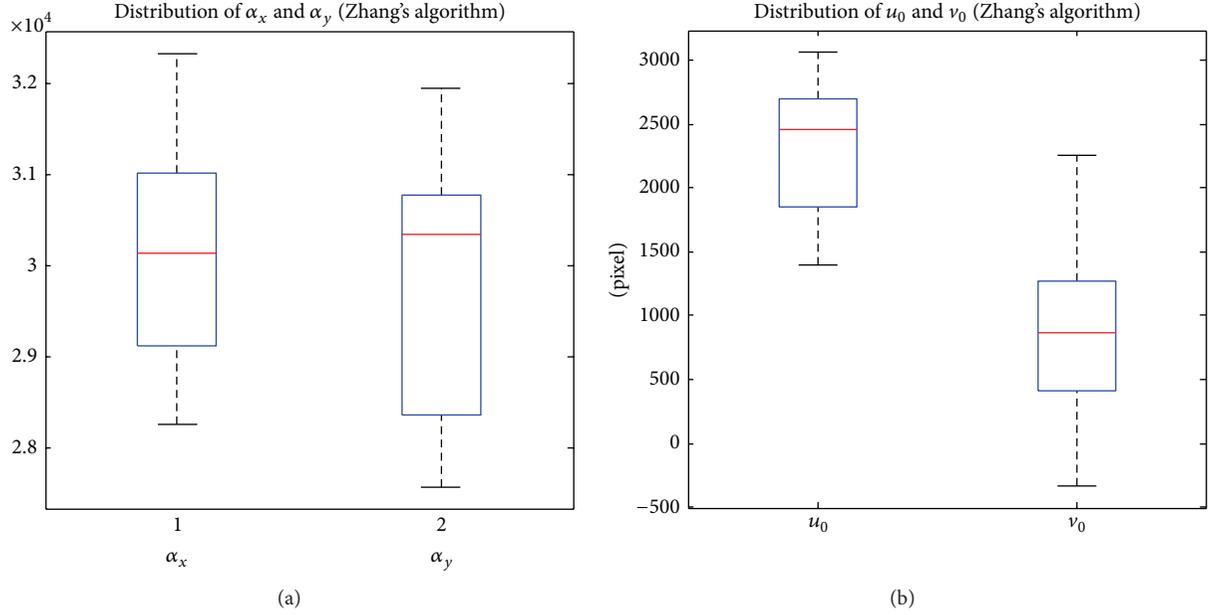


FIGURE 5: Uncertainties of α_x , α_y , u_0 , and v_0 using Zhang's algorithm [8].

the corresponding homography, it is easy to find out that the uncertainties of extrinsic parameters also depend on focal length.

One might think that it is possible to reduce the uncertainties by choosing the affine camera model. The intrinsic matrix in the affine camera model could contain less parameters (i.e., does not have a principal point), and one way to avoid over-fitting problem is to choose a simpler model. However, the scale factors still exist in the intrinsic matrix of the affine camera model. The similar derivations shown in this section can be easily extended to the affine camera model. Therefore, it can be shown that estimation uncertainties using the affine camera model also increase when focal length increases.

4. Calibration Using Regularized Least Squares

When a long focal length is used, the matrix A that is used to estimate ω_v (shown in (2)) is ill conditioned. As a result, large perturbations of the intrinsic parameters can have only small changes in the error sum of squares. Since it is often difficult to obtain other data points outside the scope of the sensor that has a limited physical dimension, in this section, we apply a simple and effective prior of the image of absolute conic to reduce the uncertainties.

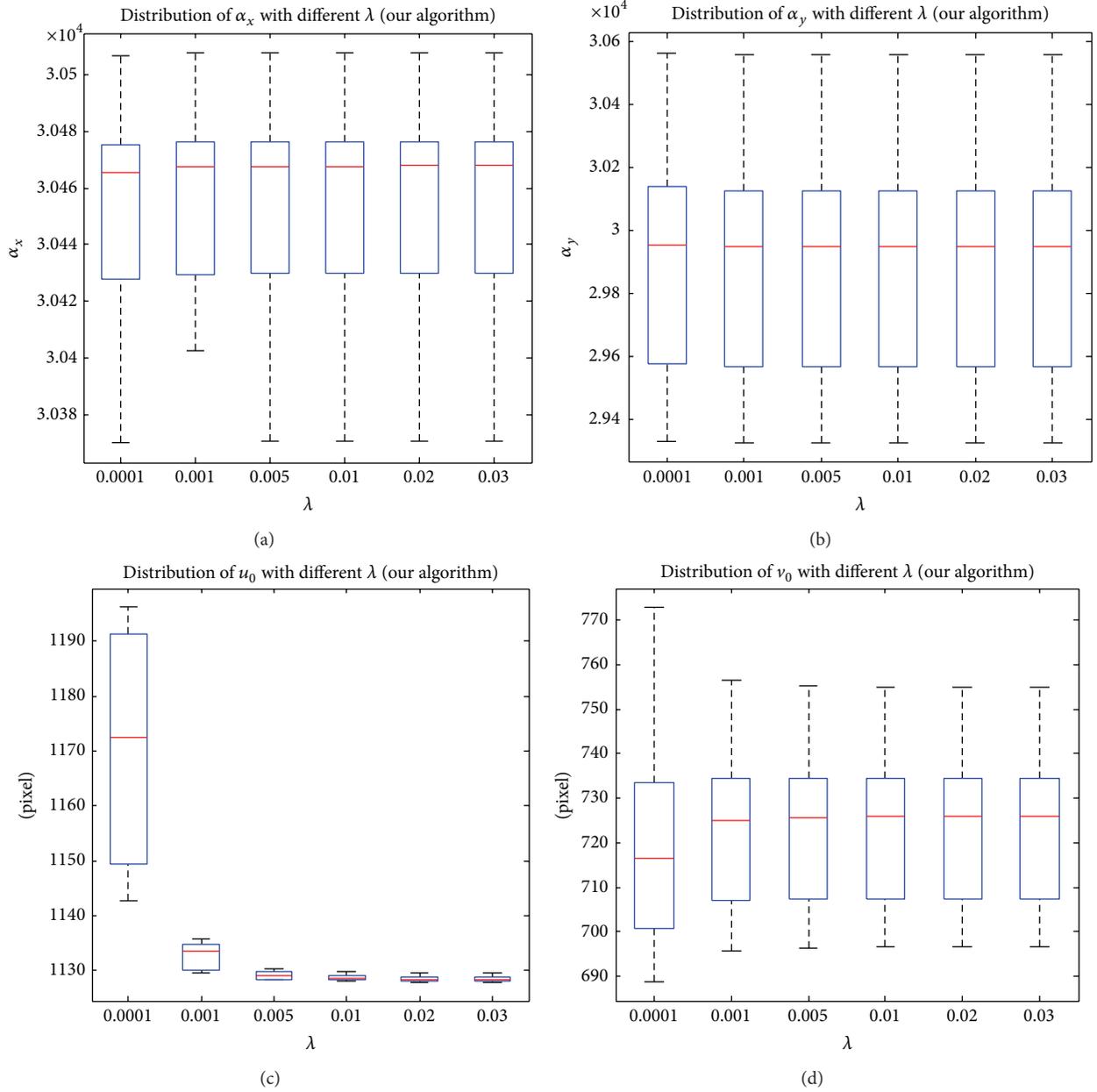
First, focal length is set as the one provided by the camera. Although this value is different from the focal length in the pin-hole camera model, they are usually in the same order. The number of pixels m_x and m_y can be computed by using sensor size and image resolution. Skew factor is close to 0. The principal point is located around the middle of an image. This location is a close approximation according to [13], which shows that the principal point varies around the image center with some nonlinear patterns when zoom and focus factors

vary. Thus, the prior knowledge of the intrinsic parameters can be denoted as $(x_1, x_2, x_3, x_4, x_5)^T = (m_x f, m_y f, 0, \bar{x}, \bar{y})^T$.

One possible solution is to apply this prior directly for the estimation of K . However, it could require a nonlinear optimization due to the Cholesky decomposition. In order to obtain a close form solution, we transform it to the prior of the image of absolute conic based on $\omega = K^{-T} K^{-1}$. Hence, the prior used in our algorithm is defined by

$$c = \begin{pmatrix} 1 \\ -\frac{x_3}{x_2} \\ \frac{x_1^2 + x_3^2}{x_2^2} \\ -\frac{x_2 x_4 - x_3 x_5}{x_2} \\ -\frac{x_5 x_1^2 + x_5 x_3^2 - x_2 x_4 x_3}{x_2^2} \\ \frac{x_1^2 x_2^2 + x_1^2 x_5^2 + x_2^2 x_4^2 - 2x_2 x_3 x_4 x_5 + x_3^2 x_5^2}{x_2^2} \end{pmatrix}. \quad (10)$$

Notice that we normalize c such that the first entry of c is 1. This is different from the original constraint $\|\omega_v\| = 1$ in (2). The reason is that some entries are very close to 0 when long focal length is used and it could be numerically unstable for solving the intrinsic parameters \mathbf{k} . For example, for a 300 mm lens and around 4000×3000 image resolution, some entries of ω are in the order of 10^{-10} and some entries during the Cholesky factorization could be in the order of 10^{-20} when $\|\omega_v\| = 1$ is applied.


 FIGURE 6: Uncertainties of α_x , α_y , u_0 , and v_0 using our algorithm.

The original homogeneous system in (2) is then converted to an inhomogeneous system by applying the prior from (10):

$$\begin{aligned} \min_{\omega_v} \quad & \|B\tilde{\omega}_v - b\|, \\ \text{s.t.} \quad & \|\tilde{\omega}_v - \tilde{c}\| \leq \eta \end{aligned} \quad (11)$$

for an appropriate value of η , where $A = [bB]$ (b is the first column of A) and $\tilde{\omega}_v$ and \tilde{c} are 2-6 elements of ω_v and c , respectively. The estimate ω_v^λ can be obtained by solving corresponding unconstrained regularized least squares problem

$$\tilde{\omega}_v^\lambda = (B^T B + \lambda I)^{-1} (B^T b + \lambda \tilde{c}) \quad (12)$$

for some positive constant λ . The expectation of $\tilde{\omega}_v^\lambda$ can be computed by

$$E(\tilde{\omega}_v^\lambda) = \tilde{\omega}_v + \lambda(B^T B + \lambda I)^{-1} (\tilde{c} - \tilde{\omega}_v). \quad (13)$$

Thus the estimator from (12) is biased after introducing the prior \tilde{c} . The second term of this equation is the bias. As λ increases the bias increases, and expectation of $\tilde{\omega}_v^\lambda$ converges to \tilde{c} eventually. In order to evaluate the covariance of $\tilde{\omega}_v$, let us define the function:

$$\Phi(\mathbf{h}, \omega_v) = (B^T B + \lambda I) \tilde{\omega}_v - B^T b - \lambda \tilde{c} = 0. \quad (14)$$

Based on the implicit function theorem in [14, 15], the Jacobian $J_{\tilde{\omega}_v}$ can be approximated by

$$J_{\tilde{\omega}_v} = -\left(\frac{\partial\Phi}{\partial\tilde{\omega}_v}\right)^{-1}\frac{\partial\Phi}{\partial\mathbf{h}}, \quad (15)$$

where $(\partial\Phi/\partial\tilde{\omega}_v)^{-1}$ and $\partial\Phi/\partial\mathbf{h}$ can be computed by

$$\begin{aligned} \left(\frac{\partial\Phi}{\partial\tilde{\omega}_v}\right)^{-1} &\approx (B^T B + \lambda I)^{-1}, \\ \frac{\partial\Phi}{\partial\mathbf{h}} &= CD, \\ C &= \begin{pmatrix} \tilde{\omega}_v & & \\ & \tilde{\omega}_v & \\ & & \ddots \end{pmatrix}, \quad D = \begin{pmatrix} \frac{\partial\mathbf{b}_1}{\partial\mathbf{h}} \\ \frac{\partial\mathbf{b}_2}{\partial\mathbf{h}} \\ \vdots \end{pmatrix}, \end{aligned} \quad (16)$$

where \mathbf{b}_i is 5×1 vector of the i th row of $(B^T B + \lambda I)\tilde{\omega}_v - B^T b$. The covariance of $\tilde{\omega}_v$ is given by

$$\Sigma_{\omega_v} = (B^T B + \lambda I)^{-1} CD \Sigma_{\mathbf{h}} D^T C^T (B^T B + \lambda I)^{-1}. \quad (17)$$

Since C and D are independent from λ and only $(B^T B + \lambda I)^{-1}$ depends on the λ , we can see that the covariance Σ_{ω_v} decreases as λ increases. The larger the λ , the closer the $\tilde{\omega}_v$ is to \tilde{c} . If we consider the mean squares error,

$$\text{MSE} = \text{Bias}^2 + \text{Variance} \quad (18)$$

it is possible to select an optimal value of $\lambda > 0$ at which the mean squared error from testing set is minimized. In practice, we could divide 2D points on a planar pattern into training and testing sets and apply the cross-validation to choose the optimal λ .

5. Experiments

We tested our proposed algorithm on simulated data and real data over a large range of settings of focal lengths and image noise.

5.1. Simulations. In our simulations, image resolution is set to 2048×1536 . Sensor size is 23.6×15.8 mm and focal lengths are 50 mm, 100 mm, 200 mm, 300 mm, 400 mm, and 500 mm. Skew factor s is set to 0.009. The principal points (u_0, v_0) are set to the image center. Table 1 gives focal lengths and corresponding scale factors used in the experiments. Gaussian noise with $\mu = 0$ and $\sigma = (1, 3)$ are added to the 2D observations. Since depth of field is limited when a long focal length is used, the observed points could be easily blurred when pan and tilt angles are large. Thus, we use a large standard deviation ($\sigma = 3$) of image noise to further test the robustness of our algorithm. The planar pattern is generated randomly with different pan/tilt angles and at different depths from the camera. Angles are uniformly distributed

TABLE 1: Scale factors of the camera used in simulations.

f (mm)	50	100	200	300	400	500
α_x	4339	8678	17356	26034	34172	43390
α_y	4861	9722	19443	29163	38886	48608

TABLE 2: Priors of the intrinsic parameters.

f	Sensor size	α_x	α_y	s	u_0	v_0
300 mm	22.2×14.8 mm	30486	30486	0	1128	752

between -60 and $+60$ degrees. Foreshortening effects are not considered in the simulations. Depths are also uniformly distributed within 6 meters. As the calibration technique in [8] is widely used in the area of computer vision, we implemented this algorithm as a baseline in order to compare calibration performance between existing algorithms and our algorithm. In our simulations, we add 5% offsets to the priors of both the focal length and the principal point. We conducted 20 trials for each configuration.

Figure 2 shows a comparison of uncertainties between the close-form solutions and the solutions from MLE when $f = 300$ mm is used. Figure 2(c) shows that RMS reprojection errors are reduced by minimizing the geometric errors. However, as the cost function is not a convex function and initial guess from the close-form solution could be far away from the global minimum, the uncertainties of intrinsic parameters cannot be reduced by the nonlinear refinement as shown in Figures 2(a) and 2(b). The results are similar to other settings in Table 1. This experiment shows that the MLE can reduce RMS errors for training data points. However, it cannot reduce the uncertainties of camera parameters.

Figures 3 and 4 show the relation between focal lengths and the uncertainties of intrinsic parameters. It shows that uncertainties increase with the increase of focal length. The absolute errors of the principal point could be very large. It indicates that the estimated principal point could be very far away from the image plane for long focal lengths. Figures 3(c), 3(d), 4(c), and 4(d) show the results by using our algorithm. the uncertainties of both the focal length and the principal point are reduced to few percents. The estimated values by using our algorithm converge to the bias (i.e., 5%), which is consistent with (13). This further means that our algorithm should be mainly used for the camera with a long focal length (e.g., $f \geq 200$ mm as shown in Figures 3 and 4). When focal length is short, we need to choose either algorithm [8] or a very small λ .

5.2. Real Data. We also test our algorithm for the real data. The camera to be calibrated is a Canon EOS 450D. The sensor size is 22.2×14.8 mm. Image resolution used in the experiments is 2256×1504 . We use a 300 mm telephoto lens in the experiments. The prior of the skew factor is set to 0. The prior of the principal point is set to the image center. The α_x and α_y are computed based on the sensor size, image resolution, and focal length. Table 2 shows the priors for the intrinsic parameters.

18 images of a planar pattern with different orientations and at different depths are captured within 6 meters. 9 images of them are randomly selected every time for calibration and the same calibration procedure is repeated 20 times. Figure 5 show the calibration results using Zhang's algorithm. We can see that the uncertainties of the intrinsic parameters are very large when a 300 mm telephoto lens is used. The estimated principal point could be far away from the image plane, which in practice is not reasonable.

Figure 6 shows our calibration results with different λ . Similar to the results of simulations, the uncertainties are reduced greatly and estimated intrinsic parameters converge to the priors when λ increases.

6. Conclusion

As a camera with a telephoto lens could be used in various vision based systems, it is necessary to calibrate the camera accurately. Many existing algorithms that are designed for cameras with relatively short focal lengths could cause large uncertainties of estimated parameters even that the RMS reprojection errors of training data are small after a nonlinear optimization. In this paper, we first give a detailed error analysis that shows the relation between uncertainties and focal length. Then we propose a robust calibration algorithm based on the regularized least squares to reduce the uncertainties. Looking into future, we will apply our approach to the camera network that contains the camera with a telephoto lens in the area of remote surveillance and scene reconstruction.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported in part by US National Science Foundation Award HRD 0833184.

References

- [1] P. J. Phillips, W. T. Scruggs, A. J. O'Toole et al., "Frvt 2006 and ice 2006 large-scale results," Tech. Rep. NISTIR 7408, National Institute of Standards and Technology, Gaithersburg, Md, USA, 2007.
- [2] H. Proença, S. Filipe, R. Santos, J. Oliveira, and L. Alexandre, "The UBIRIS.v2: a database of visible wavelength iris images captured on-the-move and at-a-distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1529–1535, 2010.
- [3] G. Guo, M. Jones, and P. Beardsley, "A System for Automatic Iris Capturing," In MERL TR2005-044, 2005.
- [4] M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi, "Super-resolution from image sequence under influence of hot-air optical turbulence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [5] O. Tonet, F. Focacci, M. Piccigallo et al., "Bioinspired robotic dual-camera system for high-resolution vision," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 55–64, 2008.
- [6] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 4, no. 3, pp. 323–344, 1987.
- [7] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, UK, 1993.
- [8] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [9] M. Pollefeys, R. Koch, and L. V. Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," in *Proceedings of the IEEE 6th International Conference on Computer Vision (ICCV '98)*, pp. 90–95, Bombay, India, January 1998.
- [10] S. N. Sinha and M. Pollefeys, "Pan-tilt-zoom camera calibration and high-resolution mosaic generation," *Computer Vision and Image Understanding*, vol. 103, no. 3, pp. 170–183, 2006.
- [11] K. Strobl, W. Sepp, and G. Hirzinger, "On the issue of camera calibration with narrow angular field of view," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems, (IROS '09)*, pp. 309–315, Sant Louis, Mo, USA, October 2009.
- [12] X. Huang, J. Gao, and R. Yang, "Calibrating pan-tilt cameras with telephoto lenses," in *Proceedings of the 8th Asian Conference on Computer Vision (ACCV '07)*, pp. 127–137, Tokyo, Japan, November 2007.
- [13] R. Willson and S. Shafer, "What is the center of the image?" in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '93)*, pp. 670–671, New York, NY, USA, June 1993.
- [14] J. C. Clarke, "Modelling uncertainty: a primer," Technical report, Department of Engineering Science, University of Oxford, Oxford, UK, 1998.
- [15] D. Liebowitz, *Camera calibration and reconstruction of geometry from images [Ph.D. thesis]*, Department of Engineering Science, University of Oxford, Oxford, UK, 2001.

Research Article

A Block-Based Regularized Approach for Image Interpolation

Li Chen,^{1,2} Xiaotong Huang,^{1,2} and Jing Tian^{1,2}

¹ School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430081, China

² Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan University of Science and Technology, Wuhan 430081, China

Correspondence should be addressed to Li Chen; chenli@ieee.org

Received 1 November 2013; Accepted 27 December 2013; Published 11 February 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Li Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a new efficient algorithm for image interpolation based on regularization theory. To render a *high-resolution* (HR) image from a *low-resolution* (LR) image, classical interpolation techniques estimate the missing pixels from the surrounding pixels based on a pixel-by-pixel basis. In contrast, the proposed approach formulates the interpolation problem into the optimization of a cost function. The proposed cost function consists of a data fidelity term and regularization functional. The closed-form solution to the optimization problem is derived using the framework of constrained least squares minimization, by incorporating Kronecker product and *singular value decomposition* (SVD) to reduce the computational cost of the algorithm. The effect of regularization on the interpolation results is analyzed, and an adaptive strategy is proposed for selecting the regularization parameter. Experimental results show that the proposed approach is able to reconstruct high-fidelity HR images, while suppressing artifacts such as edge distortion and blurring, to produce superior interpolation results to that of conventional image interpolation techniques.

1. Introduction

Image interpolation aims to render a *high-resolution* (HR) image by estimating the missing pixel information from the surrounding pixels of the observed *low-resolution* (LR) image. This image processing task is commonly referred to as image interpolation (when only a single LR image is available) or superresolution (when multiple LR images are available) [1–4]. It has a wide range of applications such as computational photography and video surveillance.

The challenge of image interpolation is to enhance the image quality while suppressing artifacts such as blurring and jagged edges. Various algorithms have been developed to address image interpolation over the years. Conventional piecewise polynomial techniques, such as bicubic interpolation, assume higher-order continuity of image intensity in the spatial domain. These techniques, however, often lead to oversmoothness in the edge and textured regions. Some edge-directed approaches adjust the algorithmic parameters according to the local features. The HR local covariance coefficients are estimated from the LR counterpart based on their geometric duality [5]. For example, inverse gradient has been employed to determine the weights of bicubic

interpolation. Different edge types are identified and used to determine different interpolation strategy for local image area [6–10]. The edge-directed interpolation is tuned based on the covariance. Other methods, including fuzzy-based [11], PDE-based [12], and regression-based method [13], have also been reported in color image interpolation and superresolution. These adaptive methods typically employ heuristic reasoning to estimate parameters such as threshold values or filter weights on a pixel-by-pixel basis. Therefore, they require extra computation to determine these local parameters, and the quality of the interpolated images may vary significantly with respect to changes in these parameters. The aforementioned deterministic image interpolation approaches cannot suppress the noise or blurring incurred in the input low-resolution image.

In view of this, a block-based regularized image interpolation approach is proposed in this paper by imposing the regularization constraint on the reconstructed high-resolution image. The relationship between the HR and LR images is exploited to formulate the interpolation problem as the optimization of a cost function. The cost function consists of a data fidelity term and Tikhonov regularization functional [14]. The closed-form solution to the optimization

problem is estimated using a new framework of constrained least squares minimization, Kronecker product, and *singular value decomposition* (SVD). A key feature of the method is its computational efficiency in reconstructing high-fidelity HR image, while alleviating common artifacts encountered by other interpolation techniques. This allows the proposed method to be employed readily in the areas of digital photography, computer vision, and medical imaging, among others.

The rest of this paper is organized as follows. Section 2 provides the mathematical formulation on the image interpolation. In Section 3, the proposed regularized image interpolation approach is presented, which is further evaluated in experimental results presented in Section 4. Finally, Section 5 concludes this paper.

2. Problem Formulation

The mathematical model between the HR and LR images plays a key role in the formulation of interpolation algorithms. Due to the finite sampling grid of the sensor array, the image acquisition processes can be modeled as integration (averaging), followed by decimation. Decimation and interpolation are inverse processes. An image decimation/interpolation model is illustrated in Figure 1. The dotted squares denote the HR pixels, while the shaded solid square represents the LR pixel. The decimation factor d is 2 in this example.

From a physical viewpoint of image acquisition, the response of each sensor is proportional to the integral of the light projected onto the surface of the sensor. Therefore, the intensity of the LR pixel in Figure 1 is determined by the corresponding effective area in the HR image grids. Consider

$$\begin{aligned}
g(0,0) &= h(-1,-1)f(-1,-1) + h(-1,0)f(-1,0) \\
&\quad + h(-1,1)f(-1,1) + h(0,-1)f(0,-1) \\
&\quad + h(0,0)f(0,0) + h(0,1)f(0,1) \\
&\quad + h(1,-1)f(1,-1) + h(1,0)f(1,0) \\
&\quad + h(1,1)f(1,1),
\end{aligned} \tag{1}$$

where $h(x,y)$ is the weight that is proportional to the area of HR pixels $f(x,y)$ in the LR pixel $g(0,0)$. In this case, $h(-1,-1) = (1-\Delta x)(1-\Delta y)$ and $h(-1,0) = (1-\Delta x)$. $h(x,y)$ is a 2D integration operator, which reflects the regions of the HR grid that contribute to the formation of a single pixel in the LR grid. This process is commonly expressed as $\mathbf{h} = \mathbf{s}\boldsymbol{\varkappa}^T$, where $\mathbf{s} = \boldsymbol{\varkappa} = (1/d)[0.5 \ 1 \cdots 1 \ 0.5]_{d+1}^T$ (if d is even) or $\mathbf{s} = \boldsymbol{\varkappa} = (1/d)[1 \ 1 \cdots 1 \ 1]_d^T$ (if d is odd).

Let $g(i,j)$ and $f(i,j)$ represent the observed LR image of size $M \times M$ and the original HR image of size $dM \times dM$, respectively, where d is the decimation factor. The

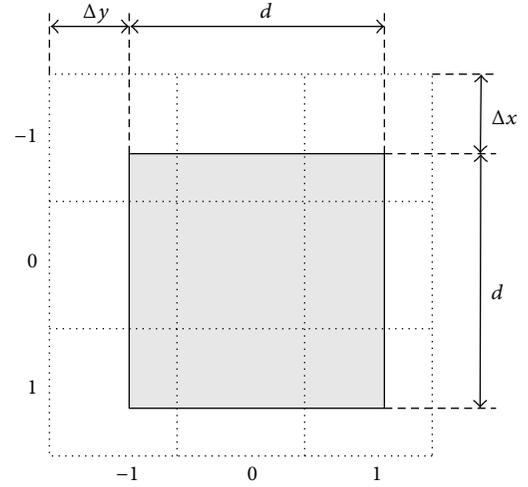


FIGURE 1: The relationship between HR and LR pixels.

relationship between the LR and HR images can be modeled as

$$\begin{aligned}
w(i,j) &= \sum_{k=-(K-1)/2}^{(K-1)/2} \sum_{l=-(K-1)/2}^{(K-1)/2} h(k,l) f(i+k, j+l) + n(i,j), \\
g(i,j) &= w(di, dj),
\end{aligned} \tag{2}$$

where $w(i,j)$ is the intermediary signal before decimation, $h(k,l)$ is a $K \times K$ two-dimensional (2D) integration operator characterizing the averaging process from the HR to LR image, and $n(i,j)$ is the additive noise. Rewriting (2) in the matrix-vector formation, we have

$$\mathbf{g} = \mathbf{D}\mathbf{H}\mathbf{f} + \mathbf{n}, \tag{3}$$

where \mathbf{f} and \mathbf{g} are the lexicographically ordered HR and LR images, respectively, \mathbf{n} is the noise vector, and \mathbf{H} and \mathbf{D} are the corresponding matrices constructed from the integration and decimation processes [15, 16]. The interpolation problem can, therefore, be formulated as solving the least squares problem for \mathbf{f} given the observation \mathbf{g} . The *Tikhonov* regularization framework is used in this paper to address this problem, as it is able to offer numerically stable and visually pleasing solution. Under this setting, \mathbf{f}_{opt} is the solution to

$$\mathbf{f}_{\text{opt}} = \min_{\mathbf{f}} \|\mathbf{D}\mathbf{H}\mathbf{f} - \mathbf{g}\|_2^2 + \lambda \|\mathbf{f}\|_2^2, \tag{4}$$

where λ is the regularization parameter which controls the relative contributions between the least square error (first term) and the regularization functional (second term). The regularization term is to ensure smoothness of the solution. The closed-form solution to the least squares problem in (4) using pseudoinverse is given by $\mathbf{f}_{\text{opt}} = [(\mathbf{D}\mathbf{H})^T(\mathbf{D}\mathbf{H}) + \lambda\mathbf{I}]^{-1}(\mathbf{D}\mathbf{H})^T\mathbf{g}$. However, the closed-form solution is impractical in many real-life applications due to the high computational cost associated with the inversion of the large matrix

$[(\mathbf{DH})^T(\mathbf{DH}) + \lambda\mathbf{I}]^{-1}$ (with a dimension of $d^2M^2 \times d^2M^2$). In view of this, we propose a computationally efficient method to address this issue in the paper.

3. Proposed Regularized Image Interpolation Approach

The formulation of the proposed interpolation algorithm involves the manipulation of Kronecker product and SVD to reduce computational cost. The regularized least square solution can achieve a good tradeoff between edge preservation and noise suppression.

3.1. Proposed New Framework of Image Interpolation through SVD and Kronecker Product. The matrix \mathbf{H} in the linear image acquisition model (3) has a block-circulant structure [15]:

$$\mathbf{H} = \begin{bmatrix} \bar{\mathbf{H}}_0 & \bar{\mathbf{H}}_{-1} & \cdots & \bar{\mathbf{H}}_1 \\ \bar{\mathbf{H}}_1 & \bar{\mathbf{H}}_0 & \cdots & \bar{\mathbf{H}}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{H}}_{-1} & \bar{\mathbf{H}}_{-2} & \cdots & \bar{\mathbf{H}}_0 \end{bmatrix}_{d^2M^2 \times d^2M^2}, \quad (5)$$

$$\bar{\mathbf{H}}_j = \begin{bmatrix} h(j,0) & h(j,-1) & \cdots & h(j,1) \\ h(j,1) & h(j,0) & \cdots & h(j,2) \\ \vdots & \vdots & \ddots & \vdots \\ h(j,-1) & h(j,-2) & \cdots & h(j,0) \end{bmatrix}_{dM \times dM},$$

where $\bar{\mathbf{H}}_j$ is the circulant matrix constructed from the 2D integration operator $h(i, j)$. In the context of image interpolation problem, the integration operator is usually characterized as a 2D separable averaging process. Therefore, the matrix \mathbf{H} can be decomposed into the Kronecker product of two matrices as

$$\mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2 = \begin{bmatrix} z_0 & z_{-1} & \cdots & z_1 \\ z_1 & z_0 & \cdots & z_2 \\ \vdots & \vdots & \ddots & \vdots \\ z_{-1} & z_{-2} & \cdots & z_0 \end{bmatrix}_{dM \times dM} \otimes \begin{bmatrix} s_0 & s_{-1} & \cdots & s_1 \\ s_1 & s_0 & \cdots & s_2 \\ \vdots & \vdots & \ddots & \vdots \\ s_{-1} & s_{-2} & \cdots & s_0 \end{bmatrix}_{dM \times dM}, \quad (6)$$

where \otimes denotes the Kronecker product. Here both \mathbf{H}_1 and \mathbf{H}_2 are circulant matrices, whose entries are constructed from the vector decomposition of integration operator: $\mathbf{h} = \mathbf{s}\boldsymbol{\varkappa}^T$, where $\mathbf{s} = [s_{-(K-1)/2}, \dots, s_0, \dots, s_{(K-1)/2}]^T$ and $\boldsymbol{\varkappa} = [z_{-(K-1)/2}, \dots, z_0, \dots, z_{(K-1)/2}]^T$.

The structure of the Kronecker product for the decimation matrix \mathbf{D} in (3) depends on the decimation factor, d . For instance, the structure of \mathbf{D} when $d = 2$ is given as

$$\mathbf{D} = \mathbf{D}_1 \otimes \mathbf{D}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{M \times dM} \otimes \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{M \times dM}. \quad (7)$$

Other decimation matrices with different decimation factors can be constructed in a similar fashion to (7) by adjusting the spacing between the consecutive 1s.

The minimization of (4) can be expressed as the constrained least squares problem via

$$\mathbf{f}_{\text{opt}} = \min_{\mathbf{f}} \left\| \begin{bmatrix} \mathbf{DH} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \quad (8)$$

Since the dimension of \mathbf{DH} is very large, it may not be computationally feasible to solve the problem directly using matrix pseudoinverse. In view of this, we develop an efficient scheme to solve (8) based on Kronecker product and SVD as follows.

First, based on (6) and (7), the Kronecker product of \mathbf{DH} can be expressed as

$$\begin{aligned} \mathbf{DH} &= (\mathbf{D}_1 \otimes \mathbf{D}_2) (\mathbf{H}_1 \otimes \mathbf{H}_2) \\ &= (\mathbf{D}_1 \mathbf{H}_1) \otimes (\mathbf{D}_2 \mathbf{H}_2). \end{aligned} \quad (9)$$

Next, we apply SVD on $\mathbf{D}_1 \mathbf{H}_1$ and $\mathbf{D}_2 \mathbf{H}_2$ to obtain

$$\begin{aligned} \mathbf{D}_1 \mathbf{H}_1 &= \mathbf{U}_1 [\boldsymbol{\Sigma}_1 | \mathbf{0}] \mathbf{V}_1^T, \\ \mathbf{D}_2 \mathbf{H}_2 &= \mathbf{U}_2 [\boldsymbol{\Sigma}_2 | \mathbf{0}] \mathbf{V}_2^T, \end{aligned} \quad (10)$$

where \mathbf{U}_1 , \mathbf{V}_1 , $\boldsymbol{\Sigma}_1$, \mathbf{U}_2 , \mathbf{V}_2 , and $\boldsymbol{\Sigma}_2$ are the standard matrices arising from SVD. Then, substituting (9) and (10) into (8), we obtain

$$\begin{aligned} \mathbf{f}_{\text{opt}} &= \min_{\mathbf{f}} \left\| \begin{bmatrix} \mathbf{DH} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \\ &= \min_{\mathbf{f}} \left\| \begin{bmatrix} \mathbf{DHf} - \mathbf{g} \\ \sqrt{\lambda}\mathbf{f} \end{bmatrix} \right\|_2^2 \\ &= \min_{\mathbf{f}} \left\| \begin{bmatrix} (\mathbf{U}_1 \otimes \mathbf{U}_2) ([\boldsymbol{\Sigma}_1 | \mathbf{0}] \otimes [\boldsymbol{\Sigma}_2 | \mathbf{0}]) \\ \times (\mathbf{V}_1 \otimes \mathbf{V}_2)^T \mathbf{f} - \mathbf{g} \\ \sqrt{\lambda}\mathbf{f} \end{bmatrix} \right\|_2^2. \end{aligned} \quad (11)$$

We further multiply the first row of the matrix by $(\mathbf{U}_1 \otimes \mathbf{U}_2)^T$ and multiply the second row of the matrix by $(\mathbf{V}_1 \otimes \mathbf{V}_2)^T$ to get

$$\mathbf{f}_{\text{opt}} = \min_{\mathbf{f}} \left\| \begin{bmatrix} (\mathbf{U}_1 \otimes \mathbf{U}_2)^T (\mathbf{U}_1 \otimes \mathbf{U}_2) \\ \times ([\boldsymbol{\Sigma}_1 \mid \mathbf{0}] \otimes [\boldsymbol{\Sigma}_2 \mid \mathbf{0}]) (\mathbf{V}_1 \otimes \mathbf{V}_2)^T \mathbf{f} \\ -(\mathbf{U}_1 \otimes \mathbf{U}_2)^T \mathbf{g} \\ \sqrt{\lambda} (\mathbf{V}_1 \otimes \mathbf{V}_2)^T \mathbf{f} \end{bmatrix} \right\|_2^2. \quad (12)$$

Finally, we denote $\mathbf{y} = (\mathbf{V}_1 \otimes \mathbf{V}_2)^T \mathbf{f}$, $\mathbf{x} = (\mathbf{U}_1 \otimes \mathbf{U}_2)^T \mathbf{g}$, to rewrite the above equation to obtain the optimal solution \mathbf{y}_{opt} as

$$\mathbf{y}_{\text{opt}} = \min_{\mathbf{y}} \left\| \begin{bmatrix} ([\boldsymbol{\Sigma}_1 \mid \mathbf{0}] \otimes [\boldsymbol{\Sigma}_2 \mid \mathbf{0}]) \\ \sqrt{\lambda} \mathbf{I} \end{bmatrix} \mathbf{y} - \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \quad (13)$$

To find out the closed-form solution of (13), we first get $\mathbf{x} = (\mathbf{U}_1 \otimes \mathbf{U}_2)^T \mathbf{g} = \text{vec}(\mathbf{U}_2^T \text{ivec}(\mathbf{g}) \mathbf{U}_1)$ and $\mathbf{f}_{\text{opt}} = (\mathbf{V}_1 \otimes \mathbf{V}_2) \mathbf{y}_{\text{opt}} = \text{vec}(\mathbf{V}_2 \text{ivec}(\mathbf{y}_{\text{opt}}) \mathbf{V}_1^T)$, since the Kronecker product for matrices \mathbf{A} , \mathbf{B} , and vector \mathbf{c} yields the property that $(\mathbf{A} \otimes \mathbf{B})\mathbf{c} = \text{vec}(\mathbf{B} \text{ivec}(\mathbf{c}) \mathbf{A}^T)$. Then, according to (13), we can get $\mathbf{y}_{\text{opt}} = \min_{\mathbf{f}} \|([\boldsymbol{\Sigma}_1 \mid \mathbf{0}] \otimes [\boldsymbol{\Sigma}_2 \mid \mathbf{0}])\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \|\mathbf{y}\|_2^2$ and use pseudoinverse as the optimal solution as

$$\mathbf{y}_{\text{opt}} = [\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I}]^{-1} \mathbf{A}^T \mathbf{x}, \quad (14)$$

where $\mathbf{A} = [\boldsymbol{\Sigma}_1 \mid \mathbf{0}] \otimes [\boldsymbol{\Sigma}_2 \mid \mathbf{0}]$, $\mathbf{A}^T = \begin{bmatrix} \boldsymbol{\Sigma}_1 \\ \mathbf{0} \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{\Sigma}_2 \\ \mathbf{0} \end{bmatrix}$, and $\mathbf{A}^T \mathbf{A} = [\boldsymbol{\Sigma}_1^2] \otimes [\boldsymbol{\Sigma}_2^2]$. Finally, we can rewrite \mathbf{y}_{opt} as

$$\begin{aligned} \mathbf{y}_{\text{opt}} &= [\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I}]^{-1} \mathbf{A}^T \mathbf{x} \\ &= \mathbf{A}^T (\mathbf{A} \mathbf{A}^T + \lambda \mathbf{I})^{-1} \mathbf{x} \\ &= \left(\begin{bmatrix} \boldsymbol{\Sigma}_1 \\ \mathbf{0} \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{\Sigma}_2 \\ \mathbf{0} \end{bmatrix} \right) \left([\boldsymbol{\Sigma}_1^2] \otimes [\boldsymbol{\Sigma}_2^2] + \lambda \mathbf{I} \right)^{-1} \mathbf{x}. \end{aligned} \quad (15)$$

By utilizing the newly derived equation (15), the computational cost of the estimation can be reduced. For an image of size $M \times M$ with decimation factor d , only $O(d^3 M^3)$ operations are needed in the new scheme as compared to $O(d^6 M^6)$ for direct implementation of pseudoinverse for (4). Therefore, this makes the algorithm computationally attractive and feasible to be implemented in real-life applications.

3.2. Regularization Analysis. After establishing the efficient algorithm to obtain the regularized solution, we study the effect of regularization on the interpolation results. The mean error and mean square error between the interpolated and original images are analyzed as follows.

The direct solution to the least square problem in (4) is given as $\mathbf{f}_{\text{opt}} = \mathbf{P}(\lambda) \mathbf{g}$, where \mathbf{f}_{opt} is the estimated HR image

and $\mathbf{P}(\lambda) = ((\mathbf{D}\mathbf{H})^T (\mathbf{D}\mathbf{H}) + \lambda \mathbf{I})^{-1} (\mathbf{D}\mathbf{H})^T$. Substituting (9) and (10) into $\mathbf{D}\mathbf{H}$ and $\mathbf{P}(\lambda)$, we obtain

$$\mathbf{D}\mathbf{H} = \sum_{i=1}^M \sum_{j=1}^M \sigma_{1,i} \sigma_{2,j} \mathbf{u}_{(i-1)M+j} \mathbf{v}_{(i-1)M+j}^T \quad (16)$$

$$\mathbf{P}(\lambda) = \sum_{i=1}^M \sum_{j=1}^M \left[\frac{\sigma_{1,i} \sigma_{2,j}}{\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda} \right] \mathbf{v}_{(i-1)M+j} \mathbf{u}_{(i-1)M+j}^T,$$

where $\boldsymbol{\Sigma}_1 = \text{diag}(\sigma_{1,i})$, $\boldsymbol{\Sigma}_2 = \text{diag}(\sigma_{2,j})$. $\mathbf{v}_{(i-1)M+j}$ and $\mathbf{u}_{(i-1)M+j}$ are the $[(i-1)dM+j]$ th and $[(i-1)M+j]$ th column vectors of the orthogonal matrices $\mathbf{V}_1 \otimes \mathbf{V}_2$ and $\mathbf{U}_1 \otimes \mathbf{U}_2$, respectively, which can be expressed as

$$\mathbf{v}_{(i-1)M+j} = \text{vec}(\tilde{\mathbf{v}}_{2,j} \tilde{\mathbf{v}}_{1,i}^T) \quad (17)$$

$$\mathbf{u}_{(i-1)M+j} = \text{vec}(\tilde{\mathbf{u}}_{2,j} \tilde{\mathbf{u}}_{1,i}^T),$$

where $\tilde{\mathbf{v}}_{1,i}$, $\tilde{\mathbf{v}}_{2,j}$, $\tilde{\mathbf{u}}_{1,i}$, and $\tilde{\mathbf{u}}_{2,j}$ are the column orthonormal basis of \mathbf{V}_1 , \mathbf{V}_2 , \mathbf{U}_1 , and \mathbf{U}_2 , respectively.

The residual error vector between the original image \mathbf{f} and the interpolation result \mathbf{f}_{opt} is equal to

$$\begin{aligned} \mathbf{r} &= \mathbf{f}_{\text{opt}} - \mathbf{f} \\ &= \left(\sum_{i=1}^M \sum_{j=1}^M \left[\frac{\sigma_{1,i}^2 \sigma_{2,j}^2}{\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda} \right] \mathbf{v}_{(i-1)M+j} \right. \\ &\quad \times \mathbf{v}_{(i-1)M+j}^T - \mathbf{I} \left. \right) \mathbf{f} \\ &\quad + \sum_{i=1}^M \sum_{j=1}^M \left[\frac{\sigma_{1,i} \sigma_{2,j}}{\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda} \right] \mathbf{v}_{(i-1)M+j} \mathbf{u}_{(i-1)M+j}^T \mathbf{n}. \end{aligned} \quad (18)$$

If the noise is zero-mean additive white Gaussian with variance σ_n^2 and independent of the image, then the expected error vector and square error between \mathbf{f}_{opt} and \mathbf{f} are given by

$$\begin{aligned} E(\mathbf{r}) &= \left(\sum_{i=1}^M \sum_{j=1}^M \left[\frac{\sigma_{1,i}^2 \sigma_{2,j}^2}{\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda} \right] \mathbf{v}_{(i-1)M+j} \right. \\ &\quad \times \mathbf{v}_{(i-1)M+j}^T - \mathbf{I} \left. \right) E(\mathbf{f}), \\ E(\|\mathbf{r}\|_2^2) &= \sigma_n^2 \sum_{i=1}^M \sum_{j=1}^M \frac{\sigma_{1,i}^2 \sigma_{2,j}^2}{(\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda)^2} \\ &\quad + E \left[\mathbf{f}^T \left(\mathbf{I} - \sum_{i=1}^M \sum_{j=1}^M \frac{\sigma_{1,i}^4 \sigma_{2,j}^4}{(\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda)^2} \right. \right. \\ &\quad \left. \left. \times \mathbf{v}_{(i-1)M+j} \mathbf{v}_{(i-1)M+j}^T \right) \mathbf{f} \right]. \end{aligned} \quad (19)$$

It can be observed from (19) that the regularization functional has biased the interpolated image, as a trade-off for numerical stability (i.e., $E(\mathbf{r}) \neq 0$ if $\lambda \neq 0$ since $\sum_{i=1}^M \sum_{j=1}^M \mathbf{v}_{(i-1)dM+j} \mathbf{v}_{(i-1)dM+j}^T = \mathbf{I}$). The *mean square error* (MSE) $E(\|\mathbf{r}\|_2^2)$ consists of two parts: the noise term and the image term. If the image is preprocessed using zero-mean centering and assumed to be white, then $E(\mathbf{f}\mathbf{f}^T) = \sigma_f^2 \mathbf{I}$, where $\sigma_f^2 = E(\mathbf{f}^T \mathbf{f})/M^2$ is the image power. This assumption is used to enable $E(\|\mathbf{r}\|_2^2)$ to be upper-bounded explicitly in terms of λ as below:

$$E(\|\mathbf{r}\|_2^2) \leq \sigma_f^2 M^2 - \sigma_f^2 \sum_{i=1}^M \sum_{j=1}^M \frac{\sigma_{1,i}^4 \sigma_{2,j}^4}{(\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda)^2} + \sigma_n^2 \sum_{i=1}^M \sum_{j=1}^M \frac{\sigma_{1,i}^2 \sigma_{2,j}^2}{(\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda)^2}. \quad (20)$$

It can be seen from (19) and (20) that regularization will reduce the impact of noise by introducing a small bias in the interpolated image.

The regularization technique is instrumental in providing satisfactory results so long as the regularization parameter is properly selected. A simple yet effective scheme to estimate the regularization parameter λ is based on the *generalized cross-validation* (GCV) function since it has been shown to be robust in other regularization schemes [16]:

$$\text{GCV}(\lambda) = \frac{\|(\mathbf{I} - \mathbf{DHP}(\lambda)) \mathbf{g}\|^2}{[\text{trace}(\mathbf{I} - \mathbf{DHP}(\lambda))]^2}. \quad (21)$$

Cross-validation is a widely used technique in the field of statistical data analysis. It uses the leave-one-out principle to address approximation of noisy data. For a fixed value of model parameter, an interpolated image is redecimated to predict the observation. GCV determines the parameter by minimizing the weighted sum of prediction errors. An advantage of GCV is that it allows the selection of the regularization parameter even when the noise power is unknown. We derive the GCV function under the context of our formulation to give

$$\text{GCV}(\lambda) = \frac{\sum_{i=1}^M \sum_{j=1}^M (\lambda x_{iM+j} / (\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda))^2}{\left(\sum_{i=1}^M \sum_{j=1}^M (\lambda / (\sigma_{1,i}^2 \sigma_{2,j}^2 + \lambda)) \right)^2}, \quad (22)$$

where $\sigma_{1,i}$ and $\sigma_{2,j}$ are the singular values of Σ_1 and Σ_2 , respectively. x_{iM+j} is the $[iM+j]$ th entry of the column vector \mathbf{x} in (15). In fact, experimental results in Section 4 show that our algorithm is robust towards different values of λ so long as it falls within a reasonable range.

4. Experimental Results

4.1. Comparison with Other Image Interpolation Approaches. The proposed approach is compared with other conventional methods: Lagrange (2nd-order polynomial) and bicubic and

TABLE 1: PSNR performance comparison of various image interpolation approaches.

Image	Noise level	Lagrange-based	Bicubic	Edge-directed	Proposed method
Lena (512 × 512)	30 dB	32.93	33.19	31.73	33.72
	40 dB	33.13	33.39	31.82	34.39
	Noiseless	33.15	33.41	31.83	34.47
Boat (512 × 512)	30 dB	30.21	30.46	29.24	31.02
	40 dB	30.33	30.58	29.30	31.43
	Noiseless	30.35	30.59	29.31	31.48
Board (256 × 256)	30 dB	14.15	14.30	13.77	14.98
	40 dB	14.17	14.31	13.79	14.99
	Noiseless	14.18	14.31	13.79	14.99
CalTrain (400 × 512)	30 dB	29.25	29.40	28.59	29.90
	40 dB	29.35	29.48	28.63	30.19
	Noiseless	29.36	29.49	28.64	30.23

edge-directed image interpolation [5]. The standard benchmark test images are used as ground truth in the experiments, including Lena (512 × 512), Boat (512 × 512), Board (256 × 256), and CalTrain (400 × 512), as shown in Figure 2. The observed low-resolution is artificially generated by filtering the original ground truth high-resolution image with an integration operator and decimating it at a rate of $d = 2$ in the horizontal and vertical directions. The downsampled image is further degraded under different noise levels to produce different SNR values: noiseless, 40 dB, and 30 dB.

The image is divided into blocks of size 16×16 with 4×4 overlapping area. There is a 4×4 overlapping in each block to avoid boundary effect. For each block, the proposed algorithm is applied to perform interpolation using (10)–(15), and the regularization parameter λ is chosen as 10^{-4} in this experiment. For performance evaluation, *peak signal-to-noise ratio* (PSNR) is used as the objective performance metric. Table 1 summarizes the results obtained for the test images using different methods. It can be observed that the proposed method achieves higher PSNR than the conventional methods consistently. On average, the proposed method offers a PSNR improvement of 0.5–1.0 dB over the Lagrange and bicubic methods and 1–2 dB over the edge-directed method. In Figures 3 and 4, the interpolated results using the Boat and Board images are given. It can be observed that the proposed method preserves the overall sharpness of the interpolated images, in particular near the edge and textured regions. The subjective human evaluation is confirmed by the objective performance measure given in Table 1.

4.2. Evaluation on the Choice of Regularization Parameter. In this section, the impact of the regularization parameter on the interpolation results is investigated. The Lena image in Figure 2(a) is selected as the test image. The same algorithm as in previous experiment is run but different values of regularization parameter λ are used, which range from 10^{-2} to 10^{-5} . In addition, the estimated λ based on the GCV function in (22) is also adopted. Table 2 summarizes the results

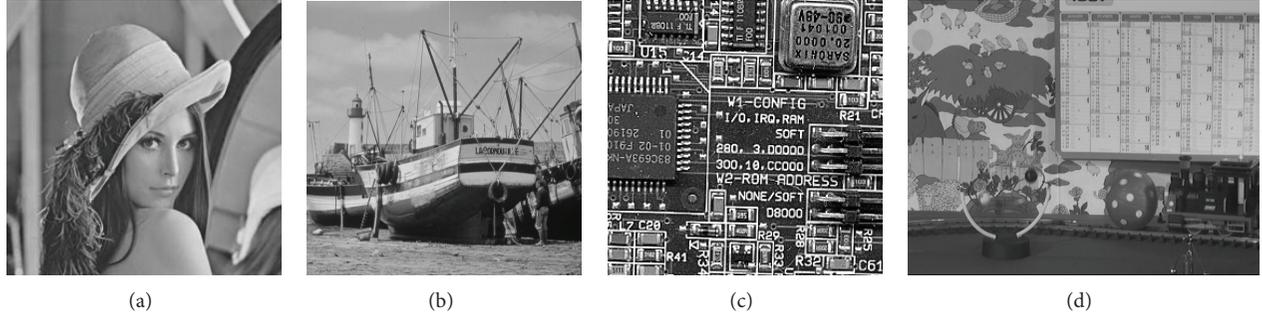


FIGURE 2: Test images used in experiments. (a) Lena, (b) Boat, (c) Board, and (d) CalTrain.

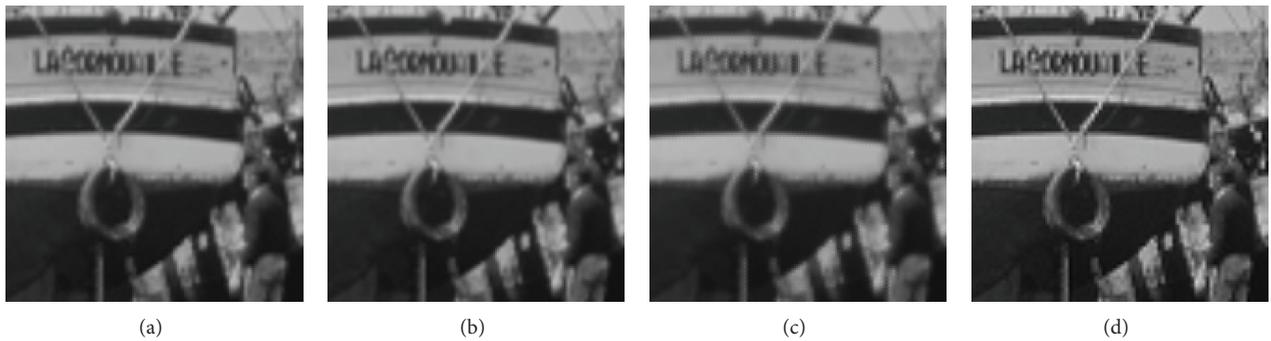


FIGURE 3: A closed-up comparison of various interpolated images (Boat image). (a) Lagrangian-based image interpolation, (b) bicubic image interpolation, (c) edge-directed image interpolation, and (d) the proposed approach.

TABLE 2: Comparison of different regularization parameters in PSNR performance.

Noise	Proposed method				GCV(λ)
	$\lambda = 10^{-2}$	$\lambda = 10^{-3}$	$\lambda = 10^{-4}$	$\lambda = 10^{-5}$	
30 dB	30.90	33.69	33.72	33.71	33.72
40 dB	31.17	34.34	34.39	34.39	34.39
Noiseless	31.21	34.42	34.47	34.47	34.47

obtained using different regularization parameters. It can be observed that this method provides consistently good results for $\lambda = 10^{-3}$ to $\lambda = 10^{-5}$ and for various noisy environments from noiseless to 30 dB noise. The estimated λ using GCV also offers comparable performance. The results suggest that the proposed algorithm is robust towards different noisy environments and regularization parameters as long as the value of λ falls within a reasonable range $[10^{-3}, 10^{-5}]$.

Note that there is an obvious improvement from $\lambda = 10^{-2}$ to $\lambda = 10^{-3}$. This gap can be explained by (20) together with the simulated *mean square error* (MSE) curve in Figure 5. The upper bound of MSE in (20) consists of three terms, where the first and third terms are positive and the second is negative. When the value of λ increases, the second term will decrease; however, the third term will increase. We simulate this situation in terms of λ ranging from 10^{-7} to 10^1 in Figure 5. The image energy is normalized to 1 (i.e., $\sigma_f^2 = 1/M^2$) and

TABLE 3: The run-time performance comparison of various image interpolation approaches.

Image	Lagrange-based	Bicubic	Edge-directed	Proposed method
Lena (512 × 512)	0.13 s	0.56 s	29.80 s	0.29 s
Boat (512 × 512)	0.13 s	0.56 s	31.80 s	0.29 s
Board (256 × 256)	0.06 s	0.20 s	6.90 s	0.15 s
CalTrain (400 × 512)	0.09 s	0.49 s	24.51 s	0.33 s

30 dB SNR noise level (i.e., $\sigma_n^2 = 10^{-3}\sigma_f^2$). Consider the ill-conditioned problem of matrix inversion; half of the singular values $\sigma_{1,i}$ are assumed to be evenly distributed in the range of $[0, 1]$ and the other half are assumed to be 0. It can be observed from Figure 5 that the magnitude of second term drops quickly and dramatically before the third term begins to increase. This leads to a basin area in the overall mean square error when λ is between a certain range.

4.3. Evaluation on Computational Complexity. These image interpolation approaches are run on a PC with Windows XP, MATLAB 7.1, CPU P4-3.4 GHz, and 1 G RAM. Each approach is run for 10 times and their average run-time is presented in Table 3, where one can see that the proposed approach

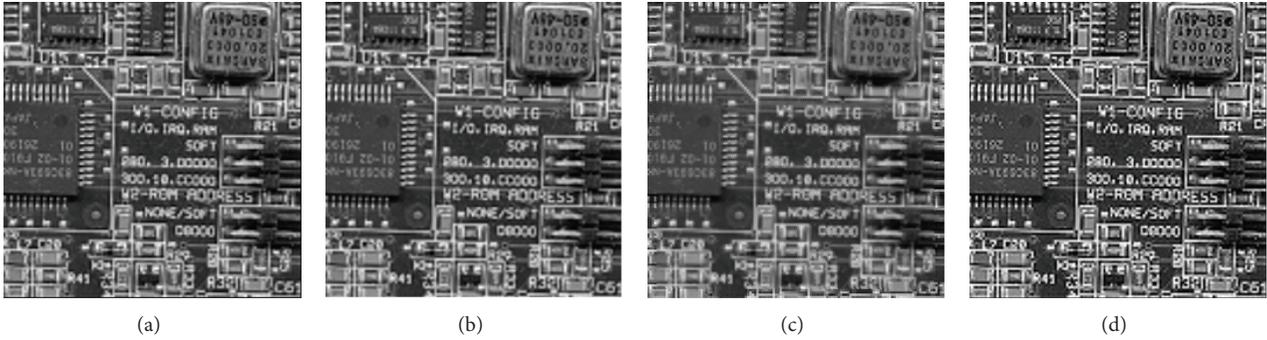


FIGURE 4: A closed-up comparison of various interpolated images (Board image). (a) Lagrangian-based image interpolation, (b) bicubic image interpolation, (c) edge-directed image interpolation, and (d) the proposed approach.

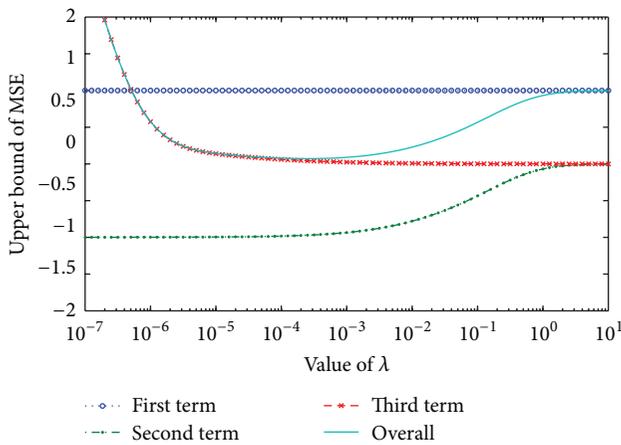


FIGURE 5: Simulation of upper bound of mean square error in terms of λ .

is much faster than the edge-directed interpolation, faster than the bicubic interpolation, but slightly slower than the Lagrange interpolation.

5. Conclusion

In this paper, a new regularization-based image interpolation algorithm using Kronecker product and singular value decomposition has been proposed. The proposed approach reduces the computational cost of interpolation while offering significant performance improvement over other conventional methods. This paper also analyzes the effect of regularization on the interpolation results and shows that the proposed approach is fairly robust towards different values of regularization parameter. It is worthwhile to further study how to extend the proposed approach for the superresolution image reconstruction.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by National Natural Science Foundation of China (nos. 61105010 and 61375017), Program for Outstanding Young Science and Technology Innovation Teams in Higher Education Institutions of Hubei Province, China (no. T201202).

References

- [1] J. Zhao, H. Feng, Z. Xu, and Q. Li, "An improved image deconvolution approach using local constraint," *Optics and Laser Technology*, vol. 44, no. 2, pp. 421–427, 2012.
- [2] J. Tian and K.-K. Ma, "A survey on super-resolution imaging," *Signal, Image and Video Processing*, vol. 5, no. 3, pp. 329–342, 2011.
- [3] P. Shanmugavadivu and K. Balasubramanian, "Particle swarm optimized multi-objective histogram equalization for image enhancement," *Optics and Laser Technology*, vol. 57, pp. 243–251, 2014.
- [4] H. Yang, Z. Zhang, M. Zhu, and H. Huang, "Edge-preserving image deconvolution with nonlocal domain transform," *Optics and Laser Technology*, vol. 54, pp. 128–136, 2013.
- [5] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.
- [6] J. W. Hwang and H. S. Lee, "Adaptive image interpolation based on local gradient features," *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 359–362, 2004.
- [7] D. D. Muresan and T. W. Parks, "Adaptively quadratic (AQua) image interpolation," *IEEE Transactions on Image Processing*, vol. 13, no. 5, pp. 690–698, 2004.
- [8] L. Wang, S. Xiang, G. Meng, H. Wu, and C. Pan, "Edge-directed single-image super-resolution via adaptive gradient magnitude self-interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 8, pp. 1289–1299, 2013.
- [9] H. Xu, G. Zhai, and X. Yang, "Single image super-resolution with detail enhancement based on local fractal analysis of gradient," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 10, pp. 1740–1754, 2013.
- [10] F. Zhou, W. Yang, and Q. Liao, "Interpolation-based image super-resolution using multisurface fitting," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3312–3318, 2012.
- [11] V. Magudeeswaran and C. G. Ravichandran, "Fuzzy logic-based histogram equalization for image contrast enhancement,"

Mathematical Problems in Engineering, vol. 2013, Article ID 891864, 10 pages, 2013.

- [12] T. Liu and Z. Xiang, "Image restoration combining the second-order and fourth-order PDEs," *Mathematical Problems in Engineering*, vol. 2013, Article ID 743891, 7 pages, 2013.
- [13] X. Liu, D. Zhao, R. Xiong, S. Ma, W. Gao, and H. Sun, "Image interpolation via regularized local linear regression," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3455–3469, 2011.
- [14] A. N. Tikhonov, *Equations of Mathematical Physics*, vol. 39, Dover Publications, New York, NY, USA, 1963.
- [15] R. L. Lagendijk and K. Biemond, *Iterative Identification and Restoration of Images*, Kluwer Academic Publishers, Boston, Mass, USA, 1990.
- [16] N. P. Galatsanos and A. K. Katsaggelos, "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation," *IEEE Transactions on Image Processing*, vol. 1, no. 3, pp. 322–336, 1992.

Research Article

An Improved Metric Learning Approach for Degraded Face Recognition

Guofeng Zou,¹ Yuanyuan Zhang,¹ Kejun Wang,²
Shuming Jiang,¹ Huisong Wan,¹ and Guixia Fu²

¹ Information Research Institute of Shandong Academy of Sciences, Jinan 250014, China

² College of Automation, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Guofeng Zou; zgf841122@163.com and Kejun Wang; wangkejun@hrbeu.edu.cn

Received 30 July 2013; Revised 20 December 2013; Accepted 20 December 2013; Published 11 February 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Guofeng Zou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To solve the matching problem of the elements in different data collections, an improved coupled metric learning approach is proposed. First, we improved the supervised locality preserving projection algorithm and added the within-class and between-class information of the improved algorithm to coupled metric learning, so a novel coupled metric learning method is proposed. Furthermore, we extended this algorithm to nonlinear space, and the kernel coupled metric learning method based on supervised locality preserving projection is proposed. In kernel coupled metric learning approach, two elements of different collections are mapped to the unified high dimensional feature space by kernel function, and then generalized metric learning is performed in this space. Experiments based on Yale and CAS-PEAL-R1 face databases demonstrate that the proposed kernel coupled approach performs better in low-resolution and fuzzy face recognition and can reduce the computing time; it is an effective metric method.

1. Introduction

The metric is a function which gives the scalar distance between two patterns. Distance metric is an important basis for similarity measure between samples, and it is one of the core issues in pattern recognition. The aim of distance metric learning is to find a distance metric matrix; its essence is to obtain another representation method with better class separability by linear or nonlinear transformation.

In recent years, some researches about distance metric have been done by researchers [1–7]. They learn a distance metric by introducing sample similarity constraint or category information; the distance metric is used to improve the data clustering or classification. These researches can be concluded to two categories: linear distance metric learning and nonlinear distance metric learning. The linear distance metric learning is equivalent to learning a linear transformation in sample space, including a variety of common linear dimensionality reduction methods, such as principal component analysis [8], linear discriminant analysis [9], and independent component analysis method [10]. The nonlinear

distance metric learning is equivalent to learning a nonlinear transformation in sample space; the locally linear embedding [11], isometric mapping [12], and Laplace mapping [13] are the traditional nonlinear methods. Recently, some new nonlinear distance metric methods have been proposed. Baghshah and Shouraki [14] proposed the nonlinear metric learning method based on pair-wise similarity and dissimilarity constraints and the geometrical structure of data. Babagholami-Mohamadabadi et al. [15] proposed the probabilistic nonlinear distance metric learning. The deep nonlinear metric learning method [16] based on neural networks is a new nonlinear metric learning method. In addition, there are some more flexible distance metric learning algorithms, which are based on kernel matrix [7, 17, 18].

These traditional distance metric learning methods are defined on the set of single attribute. If the elements belong to different sets with different attribute, these distance measurement methods are incapable for the distance metric. For example, for two images with different resolution, which can be considered to belong to different sets, obviously, the traditional distance metric method will not be able to directly

calculate the distance. The general approach is normalized operation before recognition by using interpolation algorithm or sampling algorithm. But the interpolation inevitably introduced false information, and sampling may miss some useful information, so it is difficult to get high recognition rate.

Aiming at the shortage of traditional distance metric, Li et al. proposed the coupled metric learning (CML) [19–21]. The goal of coupled metric learning is to find a coupled distance function to meet the specific requirement in given task. The essential idea is that, firstly, the data in different collections are projected to the unified coupled space, and the data should be as close as possible in this new space. Then, the generalized metric learning is performed in this unified coupled space. Obviously, this new metric method will have broader application scope and better recognition effect. However, the proposed coupled metric learning methods are based on linear transformation, which can be called linear coupled metric learning (LCML). These methods have two shortages in dealing with practical problems. First, the practical problems usually are nonlinear, and the linear transformation does not represent the features effectively. Secondly, the image needs to be converted into one-dimensional vector in the LCML algorithm; it is easy to cause the increasing of dimensions of autocorrelation matrix.

Based on the idea of coupled metric learning, we improved the supervised locality preserving projection algorithm and added supervised locality preserving information to coupled metric learning. So the improved coupled metric learning approach based on supervised locality preserving projection (SLPP-CML) was proposed. Introducing kernel technology into the coupled metric learning, we proposed the kernel coupled metric learning approach based on supervised locality preserving projection (SLPP-KCML). The SLPP-KCML realized the extension from linear coupled metric to nonlinear coupled metric. To verify the effectiveness of the proposed method, the experiments based on two face databases were performed. The experimental results show that a higher recognition rate can be achieved in the SLPP-KCML algorithm, and the operation time is reduced greatly.

2. Related Works

The traditional distance metric learning algorithm is to learn a distance function $d(x_i, x_j)$ between the data points expressed as follows:

$$d(x_i, x_j) = \|x_i - x_j\|_A = \sqrt{(x_i - x_j)^T A (x_i - x_j)}. \quad (1)$$

Distance metric learning aims to find a distance metric matrix A ; it is required that A is a real symmetric and positive semidefinite matrix; namely, $A = P^T P$, where P is a transformation matrix

$$d_A(x_i, x_j) = d_P(x_i, x_j) = \sqrt{(Px_i - Px_j)^T (Px_i - Px_j)}. \quad (2)$$

Obviously, the distance metric learning is realized by learning a transformation matrix P , so the process of distance

metric learning is equivalent to the process of obtaining other representation forms with better separability through linear or nonlinear transformation of the samples.

If $X \subset R^{D_x}$, $Y \subset R^{D_y}$ represent two different collections, respectively, the function $d(x, y)$ is the distance metric between data $x \in X$ and data $y \in Y$. If $D_x \neq D_y$, the traditional method does not work for distance metric. Even if $D_x = D_y$, because the data $x \in X$ and data $y \in Y$, which belong to different attribute collections, the distance metric has no physical meaning.

The coupled distance metric is a distance function for the data elements of different kinds of collections. The elements of collections X and Y are mapped from the original space to a common coupled space R^{D_c} by using the mapping functions f_x and f_y . Then, the distance metric is performed in the coupled space. The measured distance can be represented mathematically as

$$\begin{aligned} d^c(x, y) &= d_A(\tilde{x}, \tilde{y}) = d_A(f_x(x), f_y(y)) \\ &= \|f_x(x) - f_y(y)\|_A \\ &= \sqrt{(f_x(x) - f_y(y))^T A (f_x(x) - f_y(y))}, \end{aligned} \quad (3)$$

where matrix A is a real symmetric and positive semidefinite matrix. Letting $A = W_a W_a^T$, we can get

$$\begin{aligned} d^c(x, y) &= \sqrt{(f_x(x) - f_y(y))^T W_a W_a^T (f_x(x) - f_y(y))} \\ &= \sqrt{(W_a^T f_x(x) - W_a^T f_y(y))^T (W_a^T f_x(x) - W_a^T f_y(y))} \\ &= \|W_a^T (f_x(x) - f_y(y))\|. \end{aligned} \quad (4)$$

The goal of coupled metric learning can be achieved by minimizing the distance function; the objective function is as follows:

$$\min J = \min \sum_{(i,j) \in C} \|W_a^T (f_x(x_i) - f_y(y_j))\|^2, \quad (5)$$

where C is a correlation matrix of elements in collections X and Y . According to different supervised information, we can obtain different matrix C , so as to realize the different coupled metric learning.

3. The Coupled Metric Learning Based on Supervised Locality Preserving Projection (SLPP-CML)

The coupled distance metric learning must be used under the constraints of supervised information. In this paper, we improved the supervised locality preserving projection (SLPP) algorithm [22]. Based on the improved SLPP algorithm, we proposed the coupled metric learning method based on supervised locality preserving projection.

In order to better illustrate the coupled metric learning algorithm based on supervised locality preserving projection, we provide a theorem about the matrix norm.

Theorem 1. *Letting $A \in \mathbb{R}^{n \times n}$, then the Frobenius norm has the following properties:*

- (1) $\|A\|^2 = \text{Tr}(A^T A) = \sum_{i=1}^n \lambda_i(A^T A)$, where λ_i is the eigenvalue of the matrix $A^T A$;
- (2) $\text{Tr}(A^T A) = \text{Tr}(A A^T)$, $\text{Tr}(\cdot)$ represents the trace operation.

The coupled metric learning based on supervised locality preserving projection includes the following steps.

Step 1 (building the neighborhood relation in the same collection). We use the k nearest neighbor method. First, building within-class adjacency graph in the same collection, if the data point $x_i(y_i)$ is one of the k within-class nearest neighbors of data point $x_j(y_j)$, we connect these two data points; and then, building between-class adjacency graph in same collection; if the data point $x_i(y_i)$ is one of the k between-class nearest neighbors of data point $x_j(y_j)$, these two data points are connected.

Step 2 (building the connected relation between two collections). If the data points x_i and y_j in two different collections belong to the same class, then these two points are connected, otherwise not connected.

Step 3 (constructing the relation matrix in the same collection). According to the neighborhood relations, the relation matrixes (similarity matrixes) of within-class and between-class are constructed, respectively.

Within-class similarity matrix is W corresponding to within-class adjacency graph and the within-class similarity value is W_{ij} . The definition is as follows:

$$W_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{t}\right) & \text{if } x_i \text{ connected } x_j, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Between-class similarity matrix is B corresponding to between-class adjacency graph and the between-class similarity value is B_{ij} . It can be defined as follows:

$$B_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{t}\right) & \text{if } x_i \text{ connected } x_j, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where parameter t is the average distance between all sample points.

Step 4 (constructing the relation matrix S between two collections). The similarity value is as follows:

$$S_{ij} = \begin{cases} 1 & \text{if } x_i \text{ connected } y_j, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Step 5 (calculating the final similarity matrix C between two collections). As shown in Figure 1, the similarity relations between element $x_1 \in X$ and elements of collection Y include the following several situations.

- (a) The similarity between x_1 and y_1 : these two data points in different collections belong to the same class and they are connected to each other, so the similarity of which is $C_{11} = S_{11} = 1$.
- (b) The similarity between x_1 and y_5 : these two data points belong to different class, but the relationship between y_5 and y_3 is the between-class neighborhood relation in same collection, and the similarity B_{35} is the maximum similarity value, so similarity between x_1 and y_5 is $C_{15} = B_{35}$.
- (c) The similarity between x_1 and y_6 : these two data points belong to different class and the y_6 does not have between-class neighborhood relation with any element of collection Y of class 1. But there is a between-class neighborhood relation of same collection between y_5 and y_3 , and within-class neighborhood relation between y_5 and y_6 . So the similarity between x_1 and y_6 is defined as the product of between-class similarity B_{35} and within-class similarity W_{56} , which is the maximum similarity between y_6 and y_3 ; that is, $C_{16} = B_{35} \cdot W_{56}$.
- (d) The similarity between x_1 and y_9 : these two data points belong to different class; there are not any between-class neighborhood relations between the elements of class 1 and class 3 in collection Y ; namely, $C_{19} = 0$.

Step 6 (constructing the optimal objective function). Consider the following:

$$\begin{aligned} J &= \sum_{(i,j) \in C} \|W_a^T (f_x(x_i) - f_y(y_j))\|^2 \\ &= \sum_i \sum_j \|W_a^T (f_x(x_i) - f_y(y_j))\|^2 \cdot C_{ij}, \end{aligned} \quad (9)$$

where the functions f_x and f_y are considered to be linear; that is, $f_x(x) = W_x^T x$, $f_y(y) = W_y^T y$. The optimal objective function can be rewritten as follow:

$$\begin{aligned} J &= \sum_i \sum_j \|W_a^T (f_x(x_i) - f_y(y_j))\|^2 \cdot C_{ij} \\ &= \sum_i \sum_j \|W_a^T W_x^T x_i - W_a^T W_y^T y_j\|^2 \cdot C_{ij}. \end{aligned} \quad (10)$$

Letting $P_x = W_x W_a$, $P_y = W_y W_a$, we can get

$$\begin{aligned} J &= \sum_i \sum_j \|W_a^T W_x^T x_i - W_a^T W_y^T y_j\|^2 \cdot C_{ij} \\ &= \sum_i \sum_j \|P_x^T x_i - P_y^T y_j\|^2 \cdot C_{ij}. \end{aligned} \quad (11)$$

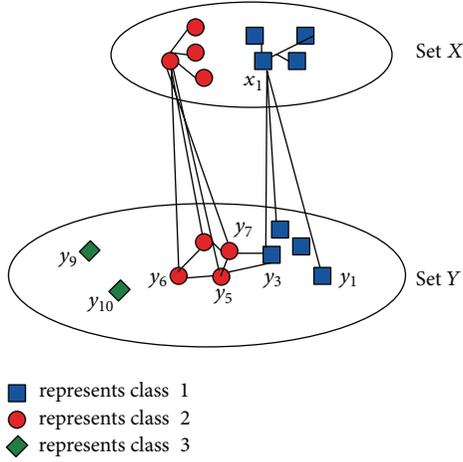


FIGURE 1: The relationship between the elements of different collections.

Therefore, our method aims to learn two linear transformations P_x and P_y .

According to Theorem 1, (12) is an alternate matrix expression of (11):

$$\begin{aligned}
 J &= \text{Tr} \left[P_x^T X F_1(C) X^T P_x + P_y^T Y F_2(C) Y^T P_y \right. \\
 &\quad \left. - P_x^T X C Y^T P_y - P_y^T Y C^T X^T P_x \right] \\
 &= \text{Tr} \left(\begin{bmatrix} P_x \\ P_y \end{bmatrix}^T \begin{bmatrix} X & Y \\ -C^T & F_2(C) \end{bmatrix} \begin{bmatrix} F_1(C) & -C \\ -C^T & F_2(C) \end{bmatrix} \begin{bmatrix} X & Y \end{bmatrix} \begin{bmatrix} P_x \\ P_y \end{bmatrix} \right), \quad (12)
 \end{aligned}$$

where $\text{Tr}(X)$ represent computing the trace of matrix X , $F_1(C)$ and $F_2(C)$ are diagonal matrixes, and their diagonal elements are the row or column sums of similarity matrix C , respectively.

Assuming that $P = \begin{bmatrix} P_x \\ P_y \end{bmatrix}$, $Z = \begin{bmatrix} X & Y \end{bmatrix}$, $\Gamma = \begin{bmatrix} F_1(C) & -C \\ -C^T & F_2(C) \end{bmatrix}$, (12) can be rewritten as follow:

$$J(P) = \text{Tr} \left[P^T \Gamma Z Z^T P \right]. \quad (13)$$

To make the equation have a unique solution, $P^T Z Z^T P = I$ and $P^T Z e = 0$ as constraints are added, where $e = [1, 1, \dots, 1]^T$ is a vector with dimensions of $(N_x + N_y) \times 1$, N_x and N_y are the numbers of samples in collections X and Y . The solution to make (13) minimized is obtained by generalized eigendecomposition of $(Z \Gamma Z^T) p = \lambda (Z Z^T) p$ and taking eigenvectors p_2, p_3, \dots, p_{m+1} corresponding to the second to $(m+1)$ th smallest eigenvalues $\lambda_2, \lambda_3, \dots, \lambda_{m+1}$. Assuming that $P = [p_2, p_3, \dots, p_{m+1}]$, its dimension is $(D_x + D_y) \times m$, D_x and D_y are the dimensions of samples in collections X and Y , so the transformation matrix P_x corresponds to the 1st to D_x th rows of P and P_y corresponds to the $(D_x + 1)$ th to D_y th rows of P .

Step 7. Bringing the matrix P_x and P_y to (11), the distance metric of the elements belonging to different collections can be realized.

4. The Kernel Coupled Metric Learning Based on Supervised Locality Preserving Projection (SLPP-KCML)

In practical dimension reduction and measurement process, the linear model is not well to represent the features, and it is difficult to map two complex collections to the same space using the linear transformation. So combining the kernel method, we extend the SLPP-CML algorithm; a nonlinear coupled metric learning methods based on the supervised locality preserving projection is proposed.

Assuming that the mapping functions f_x and f_y are nonlinear functions, namely, $f_x = \phi_x(x)$, $f_y = \phi_y(y)$, using the nonlinear mapping $\phi : R^n \rightarrow F$, $x \rightarrow \phi_x(x)$, $y \rightarrow \phi_y(y)$, the sample data can be mapped to the high dimensional Hilbert space. The criterion can be defined by

$$J = \sum_{(i,j) \in C} \left\| W_a^T \phi_x(x_i) - W_a^T \phi_y(y_j) \right\|^2. \quad (14)$$

An alternate matrix expression is as follow:

$$\begin{aligned}
 J &= \text{Tr} \left(W_a^T \Phi_x(x) G_x \Phi_x(x)^T W_a + W_a^T \Phi_y(y) G_y \Phi_y(y)^T W_a \right. \\
 &\quad \left. - W_a^T \Phi_x(x) C \Phi_y(y)^T W_a \right. \\
 &\quad \left. - W_a^T \Phi_y(y) C^T \Phi_x(x)^T W_a \right) \\
 &= \text{Tr} \left(\begin{bmatrix} W_a \\ W_a \end{bmatrix}^T \begin{bmatrix} \Phi_x(x) & \Phi_y(y) \\ \Phi_y(y) & \Phi_x(x) \end{bmatrix} \begin{bmatrix} G_x & -C \\ -C^T & G_y \end{bmatrix} \right. \\
 &\quad \left. \times \begin{bmatrix} \Phi_x(x) & \Phi_y(y) \end{bmatrix}^T \begin{bmatrix} W_a \\ W_a \end{bmatrix} \right), \quad (15)
 \end{aligned}$$

where Tr represent computing the trace of matrix, G_x and G_y are diagonal matrixes, and their diagonal elements are the row or column sums of similarity matrix C , respectively.

Letting $W_a = \Phi_x(x) A_x$, $W_a = \Phi_y(y) A_y$, we can get

$$\begin{aligned}
 J &= \text{Tr} \left(\begin{bmatrix} \Phi_x(x) A_x \\ \Phi_y(y) A_y \end{bmatrix}^T \begin{bmatrix} \Phi_x(x) & \Phi_y(y) \\ \Phi_y(y) & \Phi_x(x) \end{bmatrix} \begin{bmatrix} G_x & -C \\ -C^T & G_y \end{bmatrix} \right. \\
 &\quad \left. \times \begin{bmatrix} \Phi_x(x) & \Phi_y(y) \end{bmatrix}^T \begin{bmatrix} \Phi_x(x) A_x \\ \Phi_y(y) A_y \end{bmatrix} \right). \quad (16)
 \end{aligned}$$

The kernel function $K_{i,j}^x = (\phi(x_i) \cdot \phi(x_j))$, $K_{i,j}^y = (\phi(y_i) \cdot \phi(y_j))$; the kernel matrixes K^x and K^y are real symmetric matrices. Equation (17) is an alternative expression of (16):

$$\begin{aligned}
 J(A_x, A_y) &= \text{Tr} \left(\begin{bmatrix} A_x \\ A_y \end{bmatrix}^T \begin{bmatrix} K^x & \\ & K^y \end{bmatrix} \begin{bmatrix} G_x & -C \\ -C^T & G_y \end{bmatrix} \right. \\
 &\quad \left. \times \begin{bmatrix} K^x & \\ & K^y \end{bmatrix}^T \begin{bmatrix} A_x \\ A_y \end{bmatrix} \right). \quad (17)
 \end{aligned}$$

Obviously, the coupled metrics learning in kernel space is a process of calculating the transformation matrix A_x and A_y .

Assuming that $A = \begin{bmatrix} A_x \\ A_y \end{bmatrix}$, $K = \begin{bmatrix} K^x & \\ & K^y \end{bmatrix}$, $\Gamma = \begin{bmatrix} G_x & -C \\ -C^T & G_y \end{bmatrix}$, similar to the SLPP-CML algorithm, solving the optimal solution can be transformed into the generalized eigenvalue problem. The generalized characteristic equation is $Ea = \lambda Fa$, and $E = KTK^T$, $F = KK^T$; a is the eigenvector corresponding to eigenvalue λ . The eigenvectors corresponding to the minimum to the D_c th smallest eigenvalues construct the feature matrix A ; the size of matrix A is $(N_x + N_y) \times D_c$, where N_x and N_y are the numbers of training samples of collections X and Y . Finally, we can get the feature matrix A_x corresponding to the data matrix X and the feature matrix A_y corresponding to the data matrix Y .

In addition, the samples mapped to the high dimensional space need centering processing. In the linear coupled metric learning, the centering can be realized by abandoning the eigenvector corresponding to eigenvalue of "zero." However, the centering of nonlinear coupled metric learning in kernel space can be realized by centering the kernel matrix K^x and K^y

$$\tilde{K} = K - \frac{1}{n} \mathbf{1} \mathbf{1}_m^T - \frac{1}{n} \mathbf{1}_m \mathbf{1}^T + \frac{1}{n^2} \mathbf{1}_m \mathbf{1}_m^T, \quad (18)$$

where n is the dimension of kernel matrix K . $\mathbf{1}_m$ is a matrix with size of $n \times n$ and all elements are one.

5. Experiment and Analysis

5.1. Introduction of the Face Database. The proposed coupled metric learning approach is used for face recognition. It is tested on Yale face database [23] and CAS-PEAL-R1 face database [24]. The Yale face database contains 165 pictures of 15 people with the size of 100×100 and 256 gray levels. These images were taken in different expression and illumination conditions. In experiment, we used the former 6 images per person as training samples, a total of 90, and the other images were used as test samples.

The CAS-PEAL-R1 face database contains 30863 face images, which was divided into two parts: (1) the frontal face image subset; (2) the nonfrontal face image subset. In the experiment, we used the accessory data set of the frontal face image subsets (CAS-PEAL-R1-FRONTAL-Accessory). The face images per person in CAS-PEAL-R1-FRONTAL-Accessory contain 6 different appendages; there are 3 images with different glasses and 3 images with different hats. We selected 300 images of 50 people with the size of 360×480 and 256 gray levels in the experiment; the odd-numbered images were used as training samples and even-numbered images were used as test samples, respectively. Some training images are shown in Figure 2 and some test images are shown in Figure 3.

5.2. The Low-Resolution Face Recognition. Due to the differences between the different resolution cameras and the uncertainty of distance between camera and face, the resolution of face image that we collected is not uniform. Obviously, traditional measurement method can not be used to calculate the distance between two images with different resolution. The general handling method is interpolation operation, but the



(a) Some images in Yale face database

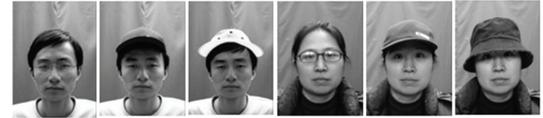


(b) Some images in CAS-PEAL-R1-FRONTAL-Accessory

FIGURE 2: Some original training face images.



(a) Some images in Yale face database



(b) Some images in CAS-PEAL-R1-FRONTAL-Accessory

FIGURE 3: Some original test face images.

interpolation operation is easy to introduce false information. With the increase in false information, the distortion degree increases, as shown in Figure 4. Aiming at the problem of recognition rate declining because of image distortion, the researchers realized the low-resolution image compensation by increasing the image restoration preprocessing. But the image restoration algorithm is more complex, and the quality of image restoration has great impact on final recognition results.

However, the proposed coupled metric learning method can directly realize the feature extraction and measurement of different resolution images. This method not only saves computing time, but also avoids the negative impact of image restoration on recognition performance. To better illustrate the experimental processes, Figure 5 gives the flow of the degraded face recognition.

In experiment, the training samples include clear face and degraded face images. The size of original normal training face image is adjusted to 64×64 pixel, and these adjusted faces are used as clear face images. However, there are not original low-resolution face images in public face database, so we obtained the low-resolution training face image through blurring and sampling original normal training face image, and the size of low-resolution face image is 16×16 .

The test samples are the low-resolution face images, which were generated through blurring and sampling original normal test face image introduced in Section 5.1.

5.2.1. Experiment 1: The Low-Resolution Face Recognition Based on SLPP-CML. Through the theoretical analysis, the

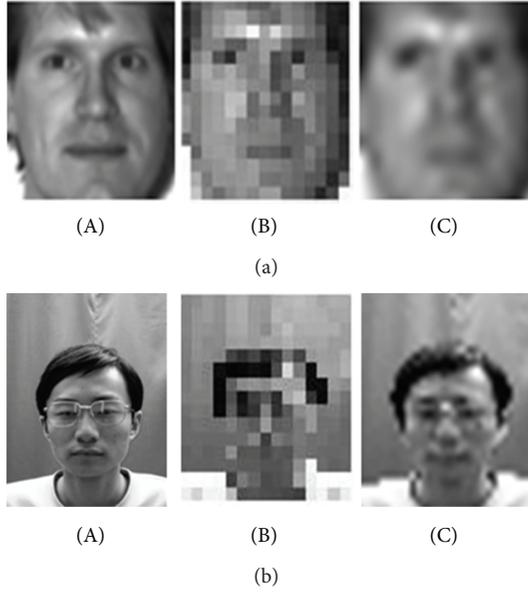


FIGURE 4: (a) (A), (B), and (C) are the normal face, low-resolution face, and face after restoration, respectively, in Yale face database. (b) (A), (B), and (C) are the normal face, low-resolution face, and face after restoration, respectively, in CAS-PEAL-R1-FRONTAL-Accessory.

SLPP-CML algorithm has two influence factors: (1) the number of neighbors k of supervised locality preserving projection; (2) the reserved dimensions D_c of the feature. Therefore, the recognition results based on different parameters should be discussed and analyzed. Figure 6 shows the change of recognition rate with the change of feature dimensions, when the number of neighbors takes different values.

These recognition rate curves are in two different face databases. These curves have a general change law, with the increase in feature dimensions; the recognition rate kept a decreasing trend after increasing, and the best recognition results can be achieved only in the optimal feature dimensions.

In Yale face database, the recognition rate kept a higher trend when feature dimensions remain 10–20. The optimal recognition rate is 86.67% when feature dimension is 10 and the number of neighbors is 5. In CAS-PEAL-R1 face database, the recognition rate can reach the maximum value 86.67%, when feature dimension is 40 and the number of neighbors is 2.

The experimental data show that the number of training samples of each class is 6 in Yale face database and the recognition effect is optimal when the number of neighbors is 5. In CAS-PEAL-R1 face database, we can obtain optimal recognition rate when the number of training samples of each class is 3 and the number of neighbors is 2. Obviously, the number of neighbors is $T - 1$; T is the number of training samples of each class.

In addition, in order to illustrate the effectiveness of SLPP-CML method, the comparative experiments were carried out. The experiment results are shown in Table 1.

The experimental data illustrated that the recognition results of feature extraction after restoration are not satisfactory. The coupled metric learning in [19] can not overcome the influences of within-class multiple modes, so the identification effect is not good. The coupled metric learning in [21] is conducive to resolving within-class multiple modes; the recognition effects have been greatly improved, but it does not fully consider the between-class relationships of training samples. The proposed SLPP-CML takes advantage of the supervisory of category information, while the within-class and between-class relationship information of training samples have been considered into the metric learning, so we can get better recognition results.

5.2.2. Experiment 2: The Low-Resolution Face Recognition Based on SLPP-KCML. The SLPP-KCML algorithm is a nonlinear coupled metric learning algorithm. Through the analysis, there are three factors which affect this algorithm: (1) the number of neighbors k of supervised locality preserving projection; (2) the reserved dimensions D_c of the feature; (3) the kernel function. Based on the experiment result, the number of the nearest neighbors is the same as that of SLPP-CML. For the kernel function, we choose the Gauss function $k(x, y) = \exp(-\|x - y\|^2/\alpha)$; the value of adjustable factor α affects the function performance. So, in this paper, the experiments were carried out according to the different adjustable factors and the change of feature dimensions on recognition rate; the experimental results are shown in Figure 7.

The curves indicated that, in Yale face database, the optimal recognition rate is 89.33% when the value of α is 0.5 and the feature dimension is 20; compared with the SLPP-CML algorithm, the recognition rate increased by 2.66%. In CAS-PEAL-R1 face database, when $\alpha = 0.7$ and the feature dimensions $D_c = 40$, recognition rate is 91.33%; compared with the SLPP-CML algorithm, the recognition rate increased by 4.66%. Obviously, the nonlinear coupled metric learning method can effectively extract the classification information of face image and obtain a high recognition rate.

Considering the training time, the SLPP-CML algorithm requires calculating the eigenvalue and eigenvector of the image covariance matrix. The resolution of clear face image is 64×64 pixel and low-resolution face image is 16×16 pixel. So the dimensions of image covariance matrix are 4352×4352 and the average training time is about 553.905 seconds.

However, the matrix of SLPP-KCML is related to the number of classes and the number of training samples of each class, so the dimension of the covariance matrix of Yale face database is 90×90 , the size of covariance matrix of CAS-PEAL-R1 face database is 150×150 , and the average training time is about 6.687 seconds. Obviously, the training speed of the SLPP-KCML algorithm is faster than SLPP-CML algorithm, and the recognition time of these two algorithms is about 0.0225 seconds. Based on the above analysis, the efficiency of SLPP-KCML algorithm is better than SLPP-CML algorithm.

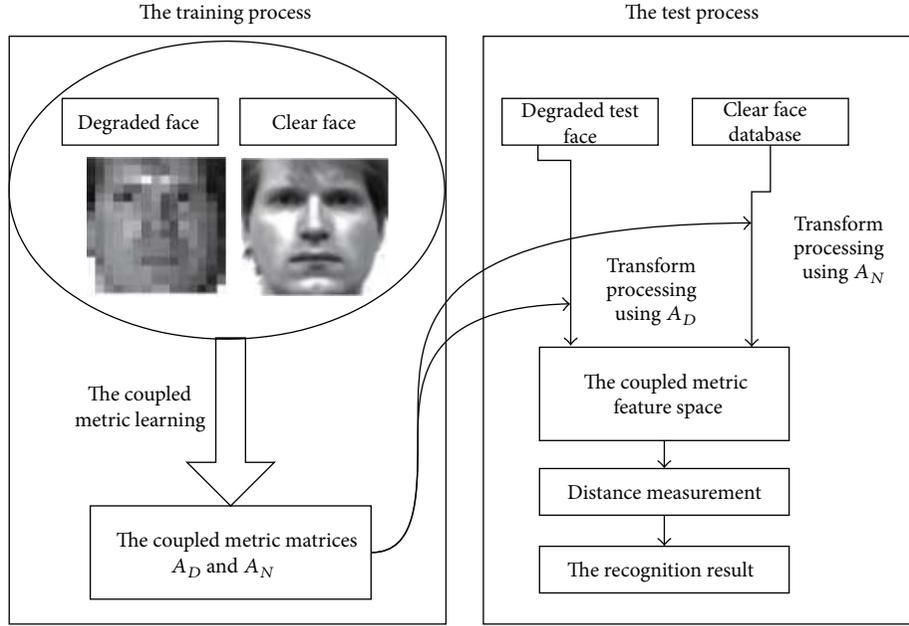


FIGURE 5: Degraded face recognition process. In training process, we obtained the matrix A_D for degraded face and the matrix A_N for clear face image. In test process, matrices A_D and A_N transformed the degraded test face and clear test face to the coupled metric space, calculating the distance.

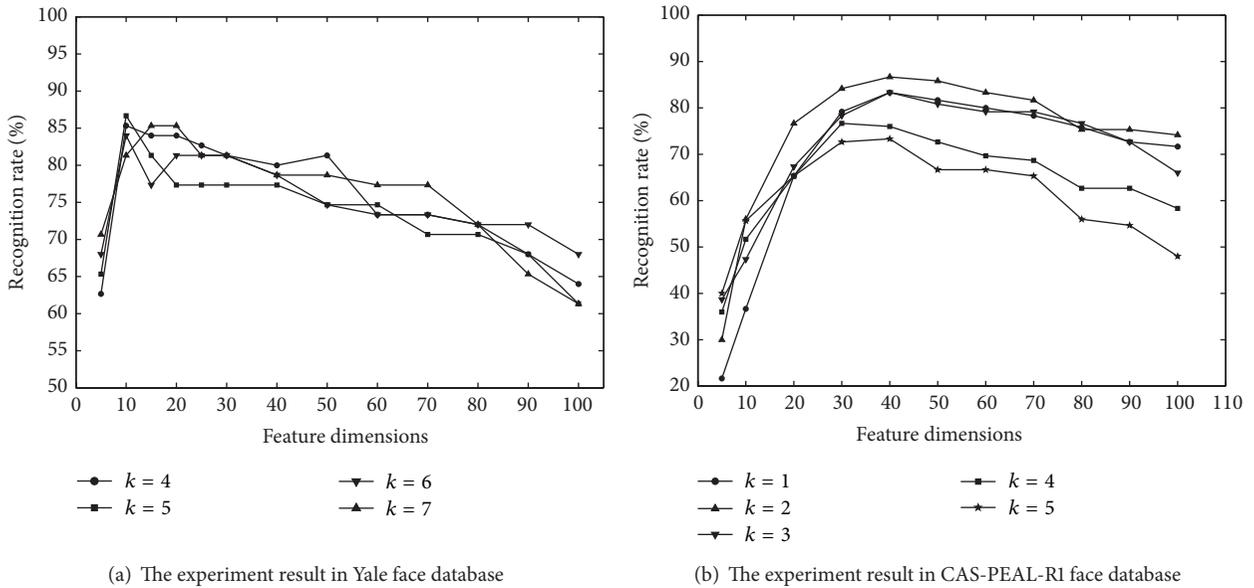
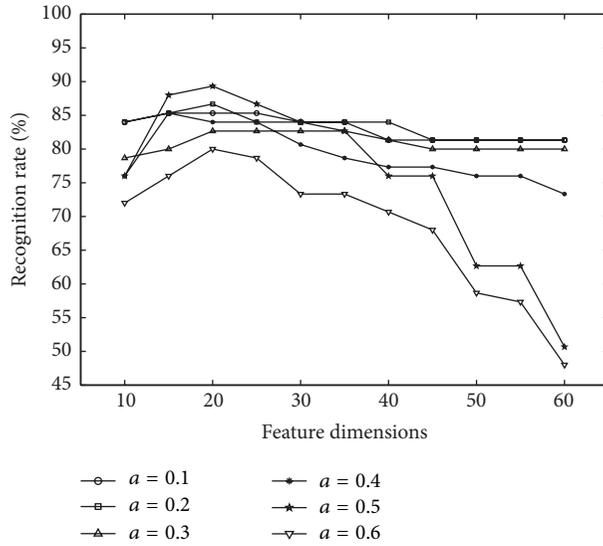


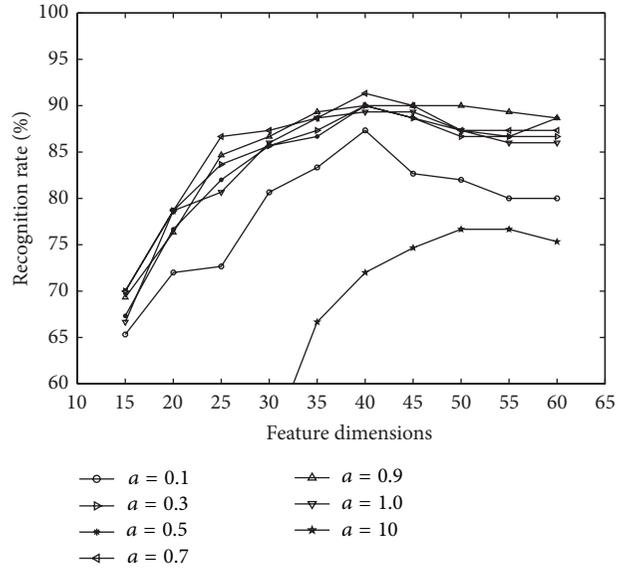
FIGURE 6: The recognition rate under different dimensions and different numbers of the nearest neighbors.

TABLE 1: Experimental comparison of this method with other methods.

Method	Yale face database	CAS-PEAL-R1 face database
Image restoration [25] + PCA	61.33	55.33
CML [19]	77.33	74.67
CLPM [21]	82.67	80.67
SLPP-CML	86.67	86.67



(a) The experiment result in Yale face database



(b) The experiment result in Yale CAS-PEAL-R1 face database

FIGURE 7: The recognition rate under different dimensions and different adjustable factors.

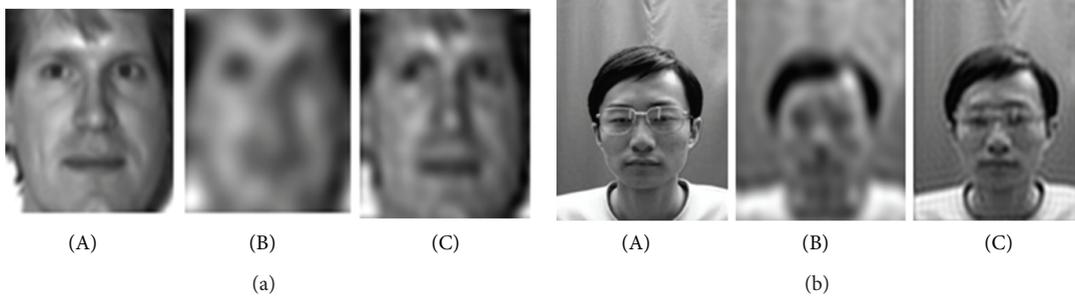


FIGURE 8: (a) (A), (B), and (C) are the normal face, blurred face image, and deblurring results, respectively, in Yale face database. (b) (A), (B), and (C) are the normal face, blurred face image, and deblurring face, respectively, in CAS-PEAL-R1-FRONTAL-Accessory.

5.3. The Fuzzy Face Recognition. Besides the low-resolution face image, the blurring image usually makes the performance of face recognition system decrease. The fuzzy face image is shown in Figure 8. Obviously, it is difficult to identify the fuzzy face image; a part of face details can be restored by deblurring algorithm, but it still cannot provide enough information in identification.

Similar to Section 5.2, we carried out the recognition experiments of SLPP-CML algorithm based on different face databases, and the comparative experiments with other methods were made. In the experiment, the clear images with size of 64×64 are the same as those used in Section 5.1, and the fuzzy face images were generated by convolution of corresponding clear face image. The training samples are composed by clear training face images and generated fuzzy face images. The test samples are the fuzzy face images by convolution of the clear test face images and the size is 64×64 pixels. The number of training and test samples has been introduced in Section 5.1. The experiment results are shown in Table 2.

The experimental data is the best recognition rate of each algorithm; the number of neighbors of SLPP-CML and SLPP-KCML algorithm is $T - 1$, where T is the number of training samples of each class. The optimal value of adjustable factor of Gaussian kernel function in SLPP-KCML algorithm is 0.7. Table 3 gives the feature dimensions of training samples in Yale and CAS-PEAL-R1 face database of SLPP-CML and SLPP-KCML algorithm.

6. Conclusions

Aiming at the problem that the traditional metric methods can not calculate the distance of the elements in different data sets, we proposed the coupled metric learning method based on supervised locality preserving projection. First, the elements of different sets are mapped to the coupled space combined with the within-class and between-class information, and then the metric matrix learning is performed. Furthermore, we extended this algorithm to nonlinear space, and the kernel coupled metric learning

TABLE 2: Experimental comparison of this method with other methods.

Method	Yale face database	CAS-PEAL-R1 face database
Deblurring [26] + PCA	64.00	68.67
CML [19]	77.33	79.33
CLPM [21]	85.67	83.33
SLPP-CML	86.67	88.67
SLPP-KCML	90.00	92.66

TABLE 3: The feature dimensions in our proposed algorithm.

Method	Yale face database	CAS-PEAL-R1 face database
SLPP-CML	15	39
SLPP-KCML	18	40

method based on supervised locality preserving projection is proposed. In kernel coupled metric learning approach, two elements of different collections are mapped to the unified high dimensional feature space by kernel function, and then the traditional metric learning is performed in this space. In order to verify the effectiveness of the proposed algorithm, we have done a lot of experiments on two face databases. This algorithm can effectively extract the face nonlinear features, and the operation is simple. Low-resolution and fuzzy face recognition experiments show that the proposed method can obtain a higher recognition rate and has a high computational efficiency.

Appendix

Proof of Theorem 1

(1) Assuming $A \in R^{n \times n}$, the Frobenius norm of matrix A is $\|A\| = (\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2)^{1/2}$. So

$$\|A\|^2 = \sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2,$$

$$A^T A = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad (\text{A.1})$$

$$= \begin{bmatrix} \sum_{i=1}^n a_{i1}^2 & \sum_{i=1}^n a_{i1}a_{i2} & \cdots & \sum_{i=1}^n a_{i1}a_{in} \\ \sum_{i=1}^n a_{i2}a_{i1} & \sum_{i=1}^n a_{i2}^2 & \cdots & \sum_{i=1}^n a_{i2}a_{in} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n a_{in}a_{i1} & \cdots & \sum_{i=1}^n a_{in}^2 \end{bmatrix}.$$

Obviously, $\text{Tr}(A^T A) = \sum_{i=1}^n a_{i1}^2 + \sum_{i=1}^n a_{i2}^2 + \cdots + \sum_{i=1}^n a_{in}^2 = \sum_{i=1}^n \sum_{j=1}^n a_{ij}^2 = \sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2$.

Therefore, $\|A\|^2 = \sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2 = \text{Tr}(A^T A)$.

According to the properties of the trace of a matrix, if $A \in R^{n \times n}$, the eigenvalue of matrix A is $\lambda_1, \lambda_2, \dots, \lambda_n$; then; $\text{Tr}(A) = \sum_{i=1}^n \lambda_i$.

Obviously, $A \in R^{n \times n}$; then, $A^T \in R^{n \times n}$, $A^T A \in R^{n \times n}$; then, $\text{Tr}(A^T A) = \sum_{i=1}^n \lambda_i(A^T A)$, where $\lambda_i(A^T A)$ is the eigenvalue of matrix $A^T A$.

Based on the above analysis, we can get that $\|A\|^2 = \text{Tr}(A^T A) = \sum_{i=1}^n \lambda_i(A^T A)$.

(2) Assuming $A \in R^{n \times m}$, $B \in R^{m \times n}$, then $AB \in R^{n \times n}$ and $BA \in R^{m \times m}$.

According to the definition of the trace, we get that $\text{Tr}(AB) = \sum_{i=1}^n (AB)_{ii}$. And then, $\text{Tr}(AB) = \sum_{i=1}^n (AB)_{ii} = \sum_{i=1}^n \sum_{j=1}^m A_{ij}B_{ji} = \sum_{j=1}^m \sum_{i=1}^n B_{ji}A_{ij} = \sum_{j=1}^m (BA)_{jj} = \text{Tr}(BA)$. The above equation shows that $\text{Tr}(AB) = \text{Tr}(BA)$; then, $\text{Tr}(A^T A) = \text{Tr}(AA^T)$.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research is supported by National Research Foundation for the Doctoral Program of Higher Education of China (20102304110004).

References

- [1] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 209–216, ACM Press, Corvallis, Ore, USA, June 2007.
- [2] S. Xiang, F. Nie, and C. Zhang, "Learning a Mahalanobis distance metric for data clustering and classification," *Pattern Recognition*, vol. 41, no. 12, pp. 3600–3612, 2008.
- [3] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.
- [4] B. Kulis, P. Jain, and K. Grauman, "Fast similarity search for learned metrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2143–2157, 2009.
- [5] H. Zhengping, L. liang, X. Chengqian et al., "Sparse distance metric learning with L1-norm constraint for one-class samples in high-dimensional space and its application," *Mathematics in Practice and Theory*, vol. 41, no. 6, pp. 116–124, 2011.
- [6] L. Wang, T. Liu, and H. Jia, "Chunk incremental distance metric learning algorithm based on manifold regularization," *Acta Electronica Sinica*, vol. 39, no. 5, pp. 1131–1135, 2011.

- [7] M. S. Baghshah and S. B. Shouraki, "Kernel-based metric learning for semi-supervised clustering," *Neurocomputing*, vol. 73, no. 7-9, pp. 1352-1361, 2010.
- [8] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417-441, 1933.
- [9] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179-188, 1936.
- [10] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287-314, 1994.
- [11] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323-2326, 2000.
- [12] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319-2323, 2000.
- [13] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in Neural Information Processing System*, vol. 14, pp. 585-591, 2002.
- [14] M. S. Baghshah and S. B. Shouraki, "Non-linear metric learning using pairwise similarity and dissimilarity constraints and the geometrical structure of data," *Pattern Recognition*, vol. 43, no. 8, pp. 2982-2992, 2010.
- [15] B. Babagholami-Mohamadabadi, A. Zarghami, H. A. Pourhaghighi, and M. T. Manzuri-Shalmani, "Probabilistic non-linear distance metric learning for constrained clustering," in *Proceedings of the 4th MultiClust Workshop on Multiple Clusterings, Multi-View Data, and Multi-Source Knowledge-Driven Clustering (MultiClust '13)*, no. 4, ACM, New York, NY, USA, 2013.
- [16] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou, "Deep nonlinear metric learning with independent subspace analysis for face verification," in *Proceedings of the 20th ACM International Conference on Multimedia*, pp. 749-752, ACM, New York, NY, USA, 2012.
- [17] M. S. Baghshah and S. B. Shouraki, "Learning low-rank kernel matrices for constrained clustering," *Neurocomputing*, vol. 74, no. 12-13, pp. 2201-2211, 2011.
- [18] S. H. Liu, J. Y. Zhang, J. Xu, and H. E. Jia, "Kernel-kNN: a new kNN algorithm based on informational energy metric," *Acta Automatica Sinica*, vol. 36, no. 12, pp. 1681-1688, 2010.
- [19] B. Li, H. Chang, S. Shan, and X. Chen, "Coupled metric learning for face recognition with degraded images," in *Advances in Machine Learning: 1st Asian Conference on Machine Learning (ACML '09)*, vol. 5828 of *Lecture Notes in Computer Science*, pp. 220-233, Springer, Halmstad, Sweden, 2009.
- [20] X. Ben, W. Meng, and R. Yan, "Dual-ellipse fitting approach for robust gait periodicity detection," *Neurocomputing*, vol. 79, pp. 173-178, 2012.
- [21] B. Li, H. Chang, S. Shan, and X. Chen, "Low-resolution face recognition via coupled locality preserving mappings," *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 20-23, 2010.
- [22] Z. Shen, Y. Pan, and S. Wang, "A supervised locality preserving projection algorithm for dimensionality reduction," *Pattern Recognition and Artificial Intelligence*, vol. 21, no. 2, pp. 233-239, 2008.
- [23] The Yale face Database, <http://cvc.cs.yale.edu/cvc/projects/yale-faces/yalefaces.html>.
- [24] The CAS-PEAL-R1 Face Database, http://www.ict.ac.cn/jszy/jsxk_zlxk/jsxk/200707/t20070706_2179538.html.
- [25] Y. Wang, Z. Sun, Z. Cai, and W. Wang, "Super-resolution image restoration based on maximum-likelihood estimation," *Chinese Journal of Scientific Instrument*, vol. 29, no. 5, pp. 949-953, 2008.
- [26] Y. Xiaofei, C. Fujie, and Y. Xuejun, "Template matching by wiener filtering," *Journal of Computer Research and Development*, vol. 37, no. 12, pp. 499-503, 2000.

Research Article

Sensor Selection and Integration to Improve Video Segmentation in Complex Environments

Adam R. Reckley,¹ Wei-Wen Hsu,¹ Chung-Hao Chen,¹ Gangfeng Ma,¹ and E-Wen Huang²

¹ Old Dominion University, Norfolk, VA 23529, USA

² National Central University, Jhongli 32001, Taiwan

Correspondence should be addressed to Chung-Hao Chen; cxchen@odu.edu

Received 1 November 2013; Accepted 28 December 2013; Published 9 February 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Adam R. Reckley et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Background subtraction is often considered to be a required stage of any video surveillance system being used to detect objects in a single frame and/or track objects across multiple frames in a video sequence. Most current state-of-the-art techniques for object detection and tracking utilize some form of background subtraction that involves developing a model of the background at a pixel, region, or frame level and designating any elements that deviate from the background model as foreground. However, most existing approaches are capable of segmenting a number of distinct components but unable to distinguish between the desired object of interest and complex, dynamic background such as moving water and high reflections. In this paper, we propose a technique to integrate spatiotemporal signatures of an object of interest from different sensing modalities into a video segmentation method in order to improve object detection and tracking in dynamic, complex scenes. Our proposed algorithm utilizes the dynamic interaction information between the object of interest and background to differentiate between mistakenly segmented components and the desired component. Experimental results on two complex data sets demonstrate that our proposed technique significantly improves the accuracy and utility of state-of-the-art video segmentation technique.

1. Introduction

Background subtraction is often considered to be a key part of any video surveillance system being used to detect objects in a single frame and/or track objects across multiple frames in a video sequence. Most of the current state-of-the-art techniques for object detection and tracking utilize some form of background subtraction that involves developing a model of the background at a pixel, region, or frame level and designating any elements that deviate from the background model as foreground. Robust object detection and tracking algorithms must be able to maintain satisfactory performance in dynamic backgrounds where the background of the image is in motion such as rippling water and illumination fluctuations. Although there are many proposed methods for object detection, a number of them achieving moderate success in dynamic backgrounds, most current state-of-the-art techniques treat objects of interest and background as separate entities ignoring any interaction. If the background of a scene

is defined as any object other than the object of interest, there are many situations when background motion affects the motion of objects of interest. For instance, the motion of an object in the ocean is going to be strongly influenced by wave action. Another example, if the object of interest is a specific vehicle, any vehicles in front of the object of interest that slow or stop will cause the object of interest to begin to slow or slow to a stop. In this paper, we hypothesize that the dynamic interaction between an object of interest and the background can provide useful information and by understanding and modeling the dynamic interaction between an object of interest and background, we can improve the performance of state-of-the-art object detection techniques. In order to test our hypothesis, we introduce heterogeneous sensing modalities to first model the dynamic interaction between an object of interest and background. Then, we select the best quality of information in terms of relevant parameters and dynamically assessing these parameters in a multisensor setting to be integrated image segmentation process.



FIGURE 1: Example of SCS-LBP algorithm performed on our data set. Image on the left is the original image and the image on the right is the result from our implementation of the SCS-LBP algorithm.

The experimental results demonstrate convincing evidence that the dynamic interaction between an object of interest and background provide valuable information that can be utilized to improve segmentation results. The structure of this paper is organized as follows: Section 3 shows the related work. The sensor selection and integration is introduced in Section 4. Section 5 illustrates the segmentation approach. Section 6 shows experimental results. Section 7 concludes this paper.

2. Motivation

At the outset of this process, we sought to detect and track an object of interest in a complex, dynamic environment. As described previously, there have been many proposed methods for video segmentation that attempt to segment objects from video sequences of dynamic environments. We chose one state-of-the-art video segmentation technique to perform segmentation on our data set of image sequences. The method that we chose to use was the spatial extended center-symmetric local binary pattern (SCS-LBP) proposed by Xue et al. [1]. When the method was utilized for segmentation on our complex data set the performance was inadequate for executing object detection and tracking. In our dataset, a water tank is used to generate dynamic backgrounds, breaking waves, high reflectance, and inconsistent motion of the object. An example of the results from the implementation by Xue et al. [1] is shown in Figure 1. As a result, Xue's method is unable to handle this case.

3. Related Work

Due to its importance in nearly all video segmentation contexts, background subtraction is a widely studied topic in computer vision and has had an abundance of literature published to its effect including surveys and evaluations of the most current and prevalent techniques [2–7]. In its most general form, background subtraction involves first creating an unambiguous model of the background in the image. Once the background model is built, the subsequent incoming frames are compared to the background model with any pixels that differ from the background model by

more than a certain threshold determined to be foreground objects. Using the determination of pixels as foreground or background, a binary foreground mask is created, completing the basic background subtraction. Robust background subtraction algorithms for real-world applications (i.e., video surveillance) must be capable of performing in dynamic environments. Rarely, in real world applications the background can be expected to remain static over an entire video sequence. The large majority of techniques presented for background subtraction focus on differing schemes for modeling the background and processes to improve current background modeling methods. In the following subsections, some popular state-of-the-art techniques are discussed.

3.1. Gaussian Mixture Model Background Subtraction. Many of the proposed techniques for background subtraction exploit the Gaussian probability density function to model the background. The parametric method proposed by Wren et al. [8] models each pixel with a single Gaussian distribution. Parametric methods assume an underlying distribution and use a set of training images that do not contain any objects of interest to estimate parameters of the underlying distribution for background modeling. Wren et al. [8] utilize the Gaussian distribution and estimate the mean background color and the covariance for each pixel in a frame. The single modal background model only provides satisfactory results when the camera and background are both static. With the exception of the simplest cases in computer vision, the assumption that both the camera and background are static is unrealistic and reduces the utility of the method in the majority of real-life situations. In order to account for slow variations in the background of an image, Stauffer and Grimson [9] proposed modeling each pixel with a mixture of Gaussians to build a background model of a sequence of images. For the purpose of this paper, slow variations in the background or slowly moving background objects can be defined as background objects whose movement in the video sequence is slower than the movement of any objects of interest in the video sequence. Using the persistence and variation of each of the Gaussians in the mixture, the Gaussians that constitute the background are determined.

Any pixel values that do not fit within any of the Gaussians that are considered part of the background are resolved to be part of a foreground object. Pixel values that are determined to be a part of the foreground are combined using connected components. The system proposed by Stauffer and Grimson [9] uses a set of K Gaussians in the mixture for modeling the background that are continuously updated based on their accuracy in modeling the background.

Due to the fact that a set number (K) of Gaussians are used in the mixture for modeling the background, during certain images, the K Gaussians may not be sufficient to automatically fully adapt the background model to the scene [10]. Zivkovic and van der Heijden [10, 11] proposed an improvement to the Gaussian mixture model to provide for the required number of components necessary to be calculated for each pixel, to allow for full adaptation to the observed scene. By using a recursive function with the weight values for each Gaussian in the mixture, Zivkovic and van der Heijden [10, 11] calculate the necessary number of components for each pixel increasing the efficiency of the system by reducing unnecessary components in the mixture of Gaussians and allowing for full adaptation of the model to the observed scene. Another proposed improvement to the Gaussian mixture model came from Lee [12] who proposed a scheme to improve the convergence rate of the Gaussian mixture model without compromising the model stability. The improved convergence rate of the Gaussian mixture model reduces the time and number of images necessary for training the algorithm. However, even with the improvements proposed by Zivkovic and van der Heijden [10, 11] and Lee [12], assuming that the pixel intensity distribution follows a Gaussian distribution may be inaccurate in dynamic scenes causing the method to fail, as stated by Xue et al. in [1].

3.2. Nonparametric Background Subtraction. In addition to the Gaussian mixture model for background subtraction, many authors proposed nonparametric methods for modeling the background to perform background subtraction. Nonparametric methods make no assumption of the underlying distribution in each pixel, instead relying on previous samples from the data to perform the background modeling. Elgammal et al. [13, 14] propose a kernel density estimation algorithm that models the background by estimating the probability of observing pixel intensity values based on a sample of previous intensity values for each pixel. Essentially, the algorithm is “estimating the probability density function by averaging the effect of a set of kernel function centered at each data point” [13]. Once the background model is estimated, background subtraction is performed. After background subtraction is completed to locate foreground objects, Elgammal et al. [13, 14] build a representation of the foreground areas to aid in the tracking of the objects and resolving any object occlusion. The kernel density estimation technique proposed by Elgammal et al. [13, 14] performs satisfactorily when the background has slow moving variation, but performance declines with the introduction of significant background movement [15]. Sheikh and Shah [16] proposed a technique for improving the performance of the

kernel density estimation. Rather than treating each image pixel as an independent random variable, Sheikh and Shah [16] contend that useful correlation can be found in pixel intensities over spatially proximal pixels that can be exploited in order to maintain accuracy over increasingly dynamic background. Additionally, Sheikh and Shah [16] propose maintaining joint background and foreground models of each pixel that can be used competitively in a maximum a posteriori probability estimation of a Markov random field (MAP-MRF) decision framework to increase the accuracy of the foreground segmentation. Even with maintaining models for both background and foreground objects to improve segmentation performance, the method proposed by Sheikh and Shah [16] requires that foreground objects have faster movement than any of the background objects [17].

Kim et al. [18, 19] propose a different nonparametric technique for background subtraction that models the background using a quantization/clustering technique. The method proposed by Kim et al. [18, 19] takes data samples at each pixel and clusters them into a set of codewords, specifically focusing on the color and brightness information. The background subtraction is performed by calculating the color distortion of the incoming pixel from the nearest cluster. If an incoming pixel has color distortion to a codeword that is less than a set detection threshold and its brightness is within the range of that codeword, the incoming pixel is classified as background. Otherwise, the incoming pixel is classified as foreground. Kim et al. [18, 19] include adaptive background model updating during illumination changes to increase performance during slowly moving background changes. However, the method is susceptible to problems when permanent structural changes occur in the background of the image due to the fact that the codeword update method does not allow for the creation of new codewords [20].

Another popular, state-of-the-art nonparametric technique, proposed by Barnich and van Droogenbroeck [20, 21], is called the visual background extractor (VIBE). Similar to other nonparametric methods, VIBE builds an estimation of the background model using data samples. However, unlike other methods, VIBE uses a random policy to select which values to include in the estimation to build the background. Barnich and van Droogenbroeck's [20, 21] random policy method in VIBE creates a smooth exponentially decaying lifespan for the values that constitute the pixel models. Additionally, the VIBE method only requires one frame to estimate the background model and can begin detecting foreground objects in the second frame. Unlike the Gaussian mixture model and kernel density estimation techniques, which update the global probability density function or estimation with each pixel value, the VIBE technique only allows the incoming pixel values to have local influence over neighboring pixels. One of the major shortcomings with the VIBE algorithm is that it has a high false detection rate [22]. The high false detection rate is due to the fact that if there are any moving objects in the initial frame from which the model is built, the moving objects will be incorporated into the background model creating a ghost region in subsequent frames [23]. Li et al. [22] proposed an improvement to the VIBE algorithm to address the problem of high false detection

rates. Li et al. [22] improve the VIBE algorithm using an adjacent frame difference algorithm which takes into account the time domain correlation between frames prior to the video. Using the time domain correlation, the improved VIBE algorithm aims to quickly remove the ghost regions in the model. The VIBE algorithm updates the model over time and would eventually eliminate the ghost, but the aim of Li et al. [22] is to remove the ghost much more quickly. Another improvement to the VIBE algorithm was proposed by van Droogenbroeck and Paquot in [24]. van Droogenbroeck and Paquot [24] improved the VIBE technique by removing foreground blobs with areas smaller or equal to 10 pixels and filling holes in the foreground objects with an area smaller or equal to 20 pixels. Additionally, van Droogenbroeck and Paquot [24] utilized the distance measure proposed by Kim et al. [19] to calculate a color distortion to improve performance of the system, upgrading from the simpler Euclidean distance measure that was used previously in VIBE. Using the new color distortion measure in conjunction with an adaptive threshold, van Droogenbroeck and Paquot [24] were able to significantly improve the performance of VIBE. However, the VIBE technique fails to reach the same level of performance on dynamic backgrounds as the Gaussian models described earlier [4]. Recall that Gaussian mixture models could perform well on dynamic background with only slowly moving background objects.

3.3. Local Binary Pattern Background Subtraction. With the exception of the VIBE algorithm, all the techniques described in the previous sections update the background model over the global probability density function or estimation. However, even the VIBE algorithm, as well as all other previously described methods, treats each incoming pixel value as independent for the purpose of creating a background model. Based on these shortcomings, the background of the image must be assumed to be static or nearly static with only slowly moving background objects for these methods to perform adequately, which prevents them from working well when attempting to detect moving objects in dynamic scenes. Comparable to the idea in the VIBE algorithm that each pixel should only affect its neighbors, Heikkilä and Pietikäinen [25] proposed a texture-based method that models each pixel as a group of adaptive local binary pattern (LBP) histograms. The local binary patterns are calculated by thresholding a P number of neighbors of a center pixel with the result being the binary pattern. This is performed for each pixel in a structure element. All of the binary patterns are placed together to form a LBP histogram for the center pixel. Background subtraction is performed by comparing the histogram for the incoming pixel against the background histograms using a histogram intersection proximity measure. If the proximity is calculated to be higher than a user-defined threshold for at least one of the background histograms, the pixel is classified as background. Otherwise, the pixel is classified as foreground.

Although the local binary pattern method proposed by Heikkilä and Pietikäinen [25] is robust to monotonic gray-scale changes and very fast to compute [25], it still does not

work adequately in dynamic scenes with intense background variations [26]. In order to address this shortcoming, Zhang et al. [26] offered an improvement to the local binary pattern technique that extended the local binary patterns from the spatial domain to the spatiotemporal domain and included an online dynamic texture extraction operator. The results from Heikkilä and Pietikäinen [25] showed improved results over the mixture of Gaussian method and kernel density estimation method; however a direct comparison to the original local binary pattern was never presented in [25]. Additionally, the inclusion of temporal information in the local binary pattern caused the method proposed by Heikkilä and Pietikäinen [25] to increase the computational load. However, even with the improvement of Heikkilä and Pietikäinen [25] to make LBP robust against local illumination changes, it struggles with uniform foreground objects in a uniform background [27].

Another improvement to the local binary pattern was introduced by Heikkilä et al. in [28]. Heikkilä et al. [28] present a new texture feature that simplifies the original local binary pattern by modifying the scheme used for comparing the neighboring pixels with the center pixel. Rather than comparing each neighboring pixel with the center pixel, Heikkilä et al. [28] compare center-symmetric pairs of pixels reducing the number of comparisons by half. Heikkilä et al. [28] incorporate their new texture feature, called center-symmetric local binary pattern (CS-LBP), into the SIFT descriptor to improve performance. The SIFT descriptor originally proposed by Lowe [29] is based on the idea that the appearance of an interest region can be characterized by the distribution of its local features. The SIFT descriptor is a 3D histogram that uses gradient as the local feature. The proposed CS-LBP feature reduced the number of required histograms increasing the computational simplicity, while providing a tolerance to illumination changes and is more robust to noise than the original local binary pattern. Although Heikkilä et al. [28] improved the LBP pattern; their improvements only utilized spatial information without taking into account temporal information [1]. Xue et al. [1] extended the CS-LBP operator from the spatial domain to the spatiotemporal domain which was designated spatial extended center-symmetric local binary pattern (SCS-LBP). SCS-LBP is capable of extracting spatial and temporal information simultaneously increasing accuracy of detection in dynamic scenes while sustaining low computational complexity by utilizing the center-symmetric scheme. A limitation of the local binary pattern and its subsequent improvements is that they are not efficient when handling large flat regions in an image (i.e., sky) due to the fact that the gray values of the neighboring pixels are very close to the value of the center pixel [25]. Chua et al. [27] proposed yet another improvement to rectify this limitation using local color features that can be represented as a local color pattern (LCP). Chua et al. [27] presented a technique that incorporates both the local binary pattern and the local color pattern to handle both texture rich areas for which the local binary pattern is effective and uniform regions where local color pattern is more effective. Using a fuzzy rule-based system, a weight is assigned and updated to the color and

texture features based on a pixel's local properties, specifically the current pixel's texture similarity score, uniformity of the binary pattern, color similarity score, and saturation value. The assigned weights are then utilized to select which features (color or texture) should be used for modeling the background at each location.

Based on the accuracy data and experimental images provided by Chua et al. [27] over nine video sequences, the method proposed by Chua et al. [27] appears to be one of the most accurate methods that were discovered during our research. Therefore, for the purpose of our experimentation, we chose to use the method proposed by Chua et al. [27] as the base method that we sought to improve.

4. Sensor Selection and Integration

Measuring the sensor belief has been an interesting research topic due to the uncertainty and imprecision involved in the sensor-based information gathering. To determine the sensor belief, the works in [18] have found out the rate of change in successive measurements from the sensor and argued that the greater the rate of change, the lower the belief. The rate of change is obtained based on the past data, the writers have defined some fuzzy rule sets to determine the self-belief of the sensors. Elgammal et al. [13] compared the performance of one sensor with another and derived a model for calculating the belief of the sensors. The performance of a sensor is determined based on the current detection outcome that supports an activity. The evidence from multiple sensors that support an activity from an abstract level is used to derive the belief value. Brutzer et al. [4] propose a dynamic belief calculation approach in the framework of a multimedia surveillance system. Using this mechanism, the belief of a set of nontrusted sensory streams evolves based on their association with other trusted streams. However, it is apparent that the determination of trusted streams would require certain precomputation, which might cause some overhead in obtaining the overall belief of the sensors and the information of interest.

4.1. Sensing Model and Measure of Uncertainty. Estimation problem is clarified using standard estimation theory. The time-dependent measurement, $\mathbf{z}_i(t)$, of sensor i with characteristics, $\lambda_i(t)$, is related to the parameters, $\mathbf{x}(t)$, that we wish to estimate through the following observation model [19]:

$$\mathbf{z}_i(t) = \mathbf{h}(\mathbf{x}(t), \lambda_i(t)), \quad (1)$$

where \mathbf{h} is a (possibly nonlinear) function depending on $\mathbf{x}(t)$ and parameterized by $\lambda_i(t)$, which represents the (possibly time dependent) knowledge about sensor i . Typical characteristics, $\lambda_i(t)$, about sensor i include sensing modality, which refers to what kind of sensor i is, sensor position \mathbf{x}_i , and other parameters, such as the noise model of sensor i and node power reserve. In (1), we consider a general form of the observation model that accounts for possibly non-linear relations between the sensor type, sensor position, noise model, and the parameters we wish to estimate. A special case of (1) would be $\mathbf{h}(\mathbf{x}(t), \lambda_i(t)) = \mathbf{f}_i(\mathbf{x}(t)) + \mathbf{w}_i(t)$, where \mathbf{f}_i

is an observation function and \mathbf{w}_i is additive, zero mean noise with known covariance. In case \mathbf{f}_i is a linear function on the parameters, (1) reduces to the linear equation:

$$\mathbf{h}(\mathbf{x}(t), \lambda_i(t)) = \mathbf{H}_i(t) \mathbf{x}(t) + \mathbf{w}_i(t). \quad (2)$$

In order to illustrate our technique, we will later consider the problem of stationary target localization with stationary sensor characteristics. Here, we assume that all sensors are acoustic sensors measuring only the amplitude of the sound signal so that the parameter vector $\mathbf{x} = [x, y]^T$ is the unknown target position, and

$$\lambda_i = [\mathbf{x}_i, \sigma_i^2]^T, \quad (3)$$

where \mathbf{x}_i is the known sensor position and σ_i^2 is the known additive noise variance. Note there is no longer a time dependence for \mathbf{x} and λ_i . Assuming that acoustic signals propagate isotropically, the parameters are related to the measurements by

$$\mathbf{z}_i = \frac{a}{\|\mathbf{x}_i - \mathbf{x}\|^{\alpha/2}} + \mathbf{w}_i, \quad (4)$$

where a is a given random variable representing the amplitude of the target, α is a known attenuation coefficient, and $\|\mathbf{x}_i - \mathbf{x}\|$ is the Euclidean norm. \mathbf{w}_i is a zero mean Gaussian random variable with variance σ_i^2 .

In the remainder of this paper, we define the *belief* as a representation of the current *a posteriori* distribution of \mathbf{x} given measurements $\mathbf{z}_1, \dots, \mathbf{z}_N : p(\mathbf{x} | \mathbf{z}_1, \dots, \mathbf{z}_N)$.

Typically, the expectation value of this distribution $\bar{\mathbf{x}} = \int \mathbf{x} p(\mathbf{x} | \mathbf{z}_1, \dots, \mathbf{z}_N) d\mathbf{x}$ is considered the estimate (i.e., the minimum mean square estimate), and we approximate the residual uncertainty by the covariance: $\Sigma = \int (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T p(\mathbf{x} | \mathbf{z}_1, \dots, \mathbf{z}_N) d\mathbf{x}$.

In order to calculate the belief based on measurements from several sensors, we must pay a cost for communicating that information. Thus, maintaining what information each sensor node has about other sensor nodes is an important decision. This is why the sensor characteristics $\lambda_i(t)$ are clearly represented because it is important to know what information is available for various information processing tasks. Since combining measurements into the belief are now assigned costs, the problem is to intelligently choose a subset of sensor measurements which provide "good" information for constructing a belief state as well as minimizing the cost of having to communicate sensor measurements to a single node. In order to choose sensors to provide "good" updates to the belief state, it is essential to understand a measure of the information.

4.2. Sensor Selection. Given the current belief state, we wish to incrementally update the belief by incorporating measurements of other nearby sensors. Among all available sensors in the network, however, not all provide useful information that improves the estimate. Furthermore, some information might be useful but redundant. The task is to select an optimal subset and to decide on an optimal order of how

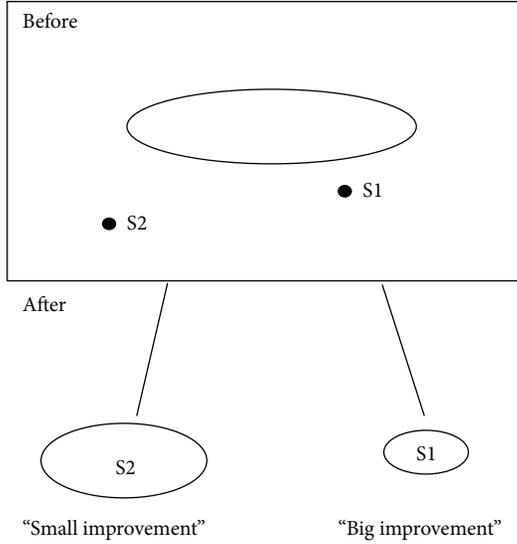


FIGURE 2: Sensor selection based on information gain of individual sensor contributions.

to incorporate these measurements into our belief update. Due to the distributed nature of the sensor network, this selection has to be done without explicit knowledge of the measurement residing at each individual sensor to avoid communicating less useful information. Hence, the decision has to be made solely based upon the sensor characteristics such as the sensor position or sensing modality and the predicted contribution of these sensors. Figure 2 shows the basic idea of optimal sensor selection. The image is based upon the assumption that estimation uncertainty can be effectively approximated by a Gaussian distribution, illustrated by uncertainty ellipsoids in the state space. In Figure 1, the solid ellipsoid indicates the belief state at time t and the dashed ellipsoids are the incrementally updated belief after incorporating an additional measurement from a sensor, S1 or S2, at the next time step.

Although in both cases, S1 and S2, the area of high uncertainty is reduced by the same amount, the residual uncertainty in the case of S2 maintains the longest principal axis of the distribution. Based on the underlying measurement task, we will choose case S1 over S2.

4.3. Measures on Expected Posterior Distribution. It is essential to define a measure of information utility to quantify the information gain provided by a sensor measurement. We want to show that information content is inversely related to the “size” of the high probability uncertainty region. We first introduce an information-theoretic definition of the utility measure. There are many kinds of measuring methods (*Covariance-Based, Fischer Information Matrix, Entropy of Estimation Uncertainty, Volume of High Probability Region, and Sensor Geometry Based Measures*) [19]. In this paper we only describe “*Expected Posterior Distribution measures*” [12] that prove to be practically useful. Our objective is to predict the information utility of a piece of nonlocal sensor data before obtaining the data. In practice, the prediction must

be based on the currently available information: the current belief state and the characteristics of the sensor of interest which includes information such as the sensor position and sensing modality that can be established beforehand. We assume there are N sensors labeled from 1 to N and the corresponding measurements of the sensors are $\mathbf{z}_1, \dots, \mathbf{z}_N$. Let $U \subset \{1, \dots, N\}$ be the set of sensors whose measurements have been incorporated into the belief. That is, the current belief is $p(\mathbf{x} | \{\mathbf{z}_i\}_{i \in U})$. The sensor selection task is to choose a sensor whose data has not been incorporated into the belief yet and which provides the most information. To be specific, let us define an information utility function φ_{Utility} that assigns a value to each probability distribution. In this case, we ignore the cost term in the objective function. The best sensor, defined by the earlier objective function, is given by

$$\hat{j} = \arg \max_{j \in V} \varphi_{\text{Utility}} \left(p(\mathbf{x} | \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\}) \right), \quad (5)$$

where V is the set of sensors whose measurements are potentially useful. The idea of using expected posterior distribution is to predict what the new belief state (posterior distribution) would look like if a simulated measurement of a sensor from the current belief state is incorporated. The utility of each sensor can then be quantified by the entropy or other measures on the new distribution from the simulated measurement. We use the tracking problem to derive an algorithm for evaluating the expected utility of a sensor. When a real new measurement is available, the new belief or posterior is evaluated using the familiar sequential Bayesian filtering [12]:

$$p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t+1)}) = C \cdot p(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)}) \cdot \int p(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}) \cdot p(\mathbf{x}^{(t)} | \mathbf{z}^{(t)}) d\mathbf{x}^{(t)}, \quad (6)$$

where $p(\mathbf{x}^{(t)} | \mathbf{z}^{(t)})$ is the current belief given a history of the measurement up to time t : $\mathbf{z}^{(t)} = \{\mathbf{z}^{(0)}, \dots, \mathbf{z}^{(t)}\}$, $p(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)})$ specifies the predefined dynamics model, $p(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)})$ is the likelihood function from the measurement of sensor j , and C is a normalization constant. How do we compute the expected value of $p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t+1)})$ without having the data $\mathbf{z}_j^{(t+1)}$ in the first place? The idea is to guess the shape of likelihood function from the current belief and the sensor position.

Without loss of generality, the current belief is represented by a discrete set of samples on a grid of the state space. This nonparametric representation of the belief state allows to represent highly non-Gaussian distribution and nonlinear dynamics. Figure 3 shows an example of the grid-based state representation. The gray squares represent the likely position of the target as specified by the current belief. The brighter the square, the more likely the target is there. For a sensor i , given the observation model $\mathbf{z}_i^{(t+1)} = h(\mathbf{x}^{(t+1)}, \mathbf{w}_i^{(t)})$, where $\mathbf{w}_i^{(t)}$ is the sensor noise, we can estimate the measurement

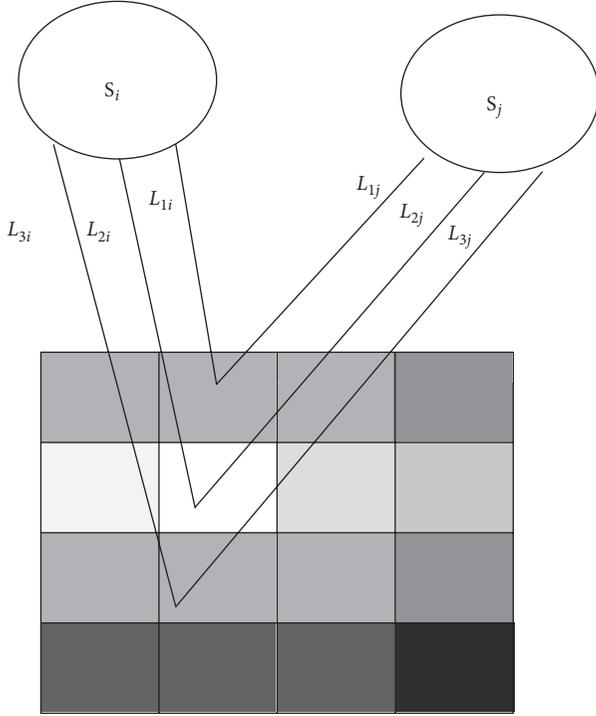


FIGURE 3: The expected likelihood function for each sensor (i or j) is a weighted sum of the marginal likelihood function conditioned at each grid in the predicted belief distribution.

$\mathbf{z}_j^{(t+1)}$ from the predicted belief and compute the expected likelihood function:

$$\begin{aligned} \hat{p}(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)}) \\ = \sum_{v_k \in S(\mathbf{x}^{(t+1)})} L_{ki}(\mathbf{x}^{(t+1)}, v_k) \times \left[p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t)}) \Big|_{\mathbf{x}^{(t+1)}=v_k} \right], \end{aligned} \quad (7)$$

where the marginal likelihood is defined as $L_{ki}(\mathbf{x}^{(t+1)}, v_k) \triangleq \hat{p}(\mathbf{z}_j^{(t+1)}(\mathbf{x}^{(t+1)} = v_k) | \mathbf{x}^{(t+1)})$ and the prediction as

$$\begin{aligned} p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t)}) \\ \triangleq \sum_{u_k \in S(\mathbf{x}^{(t)})} \left[p(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}) \Big|_{\mathbf{x}^{(t)}=u_k} \right] \\ \times \left[p(\mathbf{x}^{(t)} | \mathbf{z}^{(t)}) \Big|_{\mathbf{x}^{(t)}=u_k} \right]. \end{aligned} \quad (8)$$

Using the estimated likelihood function $\hat{p}(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)})$ from sensor i , the expected posterior belief can be obtained as follows: $\hat{p}(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t+1)}) = C \cdot \hat{p}(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)}) \cdot p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t)})$. We can then apply measures such as the entropy to the expected belief $\hat{p}(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t+1)})$, as an approximation to the true belief $p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t+1)})$. This approach can apply to non-Gaussian belief since the discrete approximation of the

belief state assumes a general form. To compute the expected belief, however, we have conditioned the expected likelihood function on the predicted belief state.

5. Object Segmentation Approach

In order to investigate the hypothesis, a method needed to be developed that incorporated the information of the dynamic interaction between the desired object of interest and the background into a state-of-the-art segmentation technique to resolve whether that information could improve the performance results of the state-of-the-art segmentation technique. However, to incorporate the dynamic interaction information for object detection and tracking, the dynamic interaction needed to be quantified in a manner that allowed it to be integrated into a state-of-the-art technique. Since the fuzzy rule-based method was able to provide a number of distinct components including the desired object of interest, in order to improve upon the performance, we sought to detect the component that corresponded to the desired object of interest from the results of the fuzzy rule-based method and remove all other components. Due to the fact that only the desired object of interest will move in a consistent, nonaccidental manner in accordance with the dynamic interaction information extracted, incorporating the dynamic interaction information will allow for detecting and tracking the desired object of interest.

In order to accomplish the detection and tracking of the desired object of interest, we developed an algorithm using dynamic interaction information that was implemented after the fuzzy rule-based method completed segmenting each frame. The dynamic information that we sought to utilize in our algorithm was the expected movement of the object of interest. Drawing inspiration from the human visual system once again, we exploited the research completed by Palmer et al. [30] that discussed spatiotemporal relatability, specifically the hypotheses of persistence and position updating. The hypothesis of persistence described by Palmer et al. [30] states that object fragments remain perceptually available for a short time in the human visual system after occlusion so that they can be integrated into later-appearing fragments. The proposed algorithm uses this insight in the way that the algorithm is implemented. Between consecutive frames in the image sequence, any detected components are maintained by the proposed algorithm. The components maintained are available for the next frame only, comparably to the way that Palmer et al. [30] describe perceptually available objects for a short period. Additionally, Palmer et al. [30] hypothesize that the human visual system maintains a representation of the velocity of an object as it moves behind occluding surfaces. Using this information, Palmer et al. [30] describe how based on the velocity of the object, even though an object may not be visible, the human visual system updates the position so that it if the object parts become visible again, it can be integrated into other visible object parts. Likewise, the proposed algorithm uses the computed dynamic movement of the desired object of interest to update the expected position of the components detected. Based upon whether

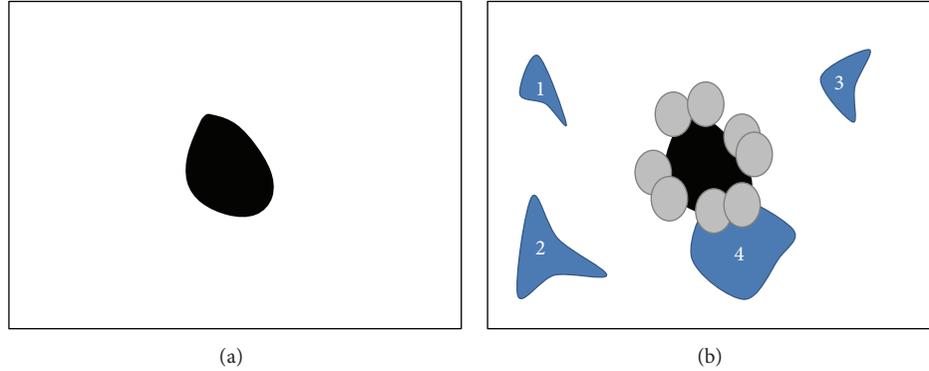


FIGURE 4: Visual representation of the algorithm process. (a) Representation of a single stored component. (b) Representation of a subsequent frame.

a detected component fulfills the updated expected position, object detection and tracking are performed.

For the first image in the sequence when the desired object of interest is detected, all of the distinct components detected by the fuzzy rule-based segmentation method are stored obeying the idea of persistence described by Palmer et al. [30]. For each component detected, every pixel included in the component, the centroid, and eight points on the extreme edge of the component were recorded. After the fuzzy rule-based method completes the segmentation on subsequent frames and detects distinct components, each of the components detected are compared against the expected motion of all of the stored components assuming that the stored components follow the motion of the desired object of interest following the concept of position updating from Palmer et al. [30]. The expected motion of the stored components is computed using the dynamic interaction information determined previously. The eight points along the edge of each of the stored components recorded earlier are at the following extreme locations: top-left, top-right, right-top, right-bottom, bottom-right, bottom-left, left-bottom, and left-top. At each of the eight locations around the extreme edges of each component, a circle with a radius corresponding to the maximum movement expected based on the dynamic interaction information calculation is positioned. Each of the positioned circles represents the possible movement of the component in each direction if the object was following the expected dynamics. In the subsequent frame, any of the components detected by the fuzzy rule-based method that are not located within the expected movement of a stored component are discarded because they do not follow the dynamic motion of the desired object of interest. In order to determine if any of the newly detected components are within the expected motion of a stored component, each of the extrema points is compared against each pixel of each newly detected component. For a single extrema point of a single stored component, the distance to every pixel in each of the newly detected components is calculated in accordance with the equation below:

$$\text{Measured Distance} = \sqrt{(x_n - x_e)^2 + (y_n - y_e)^2}, \quad (9)$$

where (x_n, y_n) are the coordinates of a pixel in one of the newly detected components and (x_e, y_e) are the coordinates of the extrema point. The measured distance is compared against the expected movement per image calculated previously. If the measured distance is less than the expected movement per image, then the newly detected component to which the pixel belongs is considered to have a presence within the expected movement of the stored component. Any detected components that have a presence within the expected movement of a stored component are accumulated for the next frame since it could possibly represent the desired object of interest. Pseudocode depicting the process for determining if a new detected component has a presence within the expected movement of a stored component is shown in Algorithm 1.

A visual representation of this process is displayed in Figure 4. The image on the left of Figure 4 shows a representation of a stored component. The image on the right of Algorithm 1 shows a visual representation of the comparison process of the algorithm. The blue components represent newly detected components in the current frame whereas the black component represents the stored component from the previous frame. Since only new detected component number four exists inside the expected movement of the object of interest (represented by the gray, transparent circles), it is the only newly detected component that is stored for the next frame. The rest of the newly detected components (1–3) would be discarded. After a series of subsequent frames, using the same process, ultimately only the desired object of interest will remain. However, in rare occurrences, the fuzzy rule-based method will mistakenly detect a component that has no directional movement but appears continuously in a majority of frames and would be retained throughout the sequence of images using the method described.

Therefore, in addition to the method described, another evaluation is performed that will establish whether or not the detected components that are retained for multiple frames maintain an overall directional movement. In addition to calculating the extreme points on the edge of stored components, the centroid of the component is also calculated and maintained. After each image where a component is maintained, the previous centroids that correspond to that

```

Determine the stored components of the previous frame
Determine the detected components of the current frames
FOR (each extreme point on the stored component)
  FOR (each pixel included in the new component detected)
    Calculate the distance ( $M_D$ ) between the current extreme pixel and the
    current pixel from the new detected component
  IF ( $M_D$  is less than the expected movement)
    (i) New detected component has presence in the expected
    movement, store the new detected component for next frame
    (ii) Break to next detected component
  END-IF
End-FOR
END-FOR

```

ALGORITHM 1: Pseudocode for determining if a new detected component falls within the expected movement of a stored component.

component are maintained, creating a register of the history of the component's centroid. Using this register of centroid data, every twenty frames, the distance the component has traveled is calculated and compared to a user-defined threshold. If the distance that the component has traveled is less than the user-defined threshold, the component is discarded in subsequent frames removing the mistakenly detected components from the fuzzy rule-based technique that have no directional movement. Figure 5 shows a visual representation of this process. For clarity, only five centroids are represented for two objects, whereas in the proposed algorithm the process is performed after twenty frames. In Figure 6, the distance traveled by the object represented by the black centroids is obviously greater than the distance traveled by the object represented by the blue centroids. Therefore, depending on the user-defined threshold, the object represented by the black centroids would be retained and stored, while the object represented by the blue centroids would be discarded.

Additionally, in order to facilitate understanding of the proposed algorithm, pseudocode of the proposed algorithm is provided in Algorithm 2. The pseudocode in Figure 5 describes the proposed algorithm from the completion of the fuzzy rule-based method proposed by Herrero and Bescos [5] until the completion of the proposed algorithm. In Algorithm 2, user-defined threshold is based on the maximum speed of the desired target and the number, 20, is based on the processed frames per second in our case.

Recall that the fuzzy rule-based method utilized two features for segmentation, the LBP (texture) and the LCP (color). Starting with the LBP, there are a number of parameters that will have an impact on the number of iterations of the algorithm. Let F represent the number of frames where each frame size is M rows by N columns. For the purpose of extracting the LBP, an R by R structure element is used with each LBP referencing P neighboring points around the center position of the structure element. In addition to finding the LBP for the R by R structure element, the algorithm also calculates the histogram statistic of three different color spaces for each pixel in the structure element as well. The fuzzy rule-based technique will take $F * M * N * (R * R * P + R * R * 3 + K)$ iterations to complete. If we regard the parameters

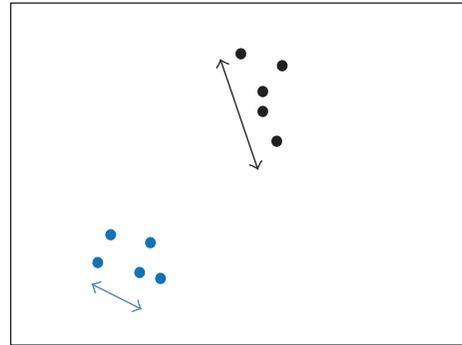


FIGURE 5: Visual representation of the algorithm process for removing objects mistakenly detected in a majority of frames with minimal directional movement.

P and K as the fixed constants, using the big O asymptotic notation, the complexity is in the $O(F * M * N * R^2)$ level.

However, in addition to the fuzzy rule-based, the proposed algorithm is performed after the completion of the fuzzy rule-based segmentation for every frame. The proposed algorithm compares the connections of the segmented components between two successive frames. Each pixel of each component in the new frame is compared against the eight extrema points from the stored components. Once we locate any connection between the current component with some previous component, we do not have to compare the current component against the rest of the stored components. Consequently, if there were C components in the previous frame, each pixel in the current component will check C times at most, and the total number of pixels from the components in the current frame that will have to be checked will be at most the entire image or M by N . Therefore, the iteration for checking the connection will take at most $M * N * C * 8$. Due to the fact that the number of components detected for each frame (C) is very limited, we can regard the number of components detected for each frame as a constant. Using these constraints, the complexity of the proposed algorithm is $O(F * M * N)$ for processing F frames. Therefore, the total complexity for the fuzzy rule-based method and the proposed

```

Determine the maximum motion dynamic expected per image in the form of pixel radius
FOR (each frame in the image sequence)
    (i) Perform fuzzy rule-based technique
    (ii) Perform connected component analysis on result of fuzzy rule-based technique
IF (current frame is the first frame in sequence to contain object of interest)
    Store all components detected by connected component analysis
    Each component stored has the following auxiliary information:
    (i) All pixels included in the component
    (ii) Centroid of the component
    (iii) Eight extrema points on the edge of the component
Else
    FOR (each new component detected)
        FOR (each of the sorted components from previous frame)
            FOR (each of the extrema points of stored components)
                FOR (each pixel included in the current new component)
                    Calculate the distance from current extrema point to current
                    pixel in current new component
                IF (distance < max motion dynamic expected)
                    (i) Store current new component and auxiliary
                    information for subsequent frames
                    (ii) Break to next new component
                END-IF
            END-FOR
        END-FOR
    END-FOR
    IF (frame processed in a multiple of 20)
        Calculate the total distance travelled by the current new component
        IF (distance travelled is greater than user-defined threshold)
            Store current new component and auxiliary information for
            subsequent frames
        ELSE
            Discard the current new component
        END-IF
    END-IF
END-FOR
END-IF
END-FOR

```

ALGORITHM 2: Pseudocode of the proposed algorithm.

algorithm will be $O(F * M * N * R^2) + O(F * M * N)$, which simplifies into the $O(F * M * N * R^2)$ level.

6. Experimental Results

After the development of the improvement to the fuzzy rule-based method proposed by Chua et al. [27], the improved method was tested on three sequences of images from our data set. In order to numerically compare our technique with the results of the fuzzy rule-based technique, we utilized the same method that was used by Chua et al. [27] when evaluating their fuzzy rule-based technique in the original paper called F -measure [27]. The F -measure method used by Chua et al. [27] is defined as follows:

$$F = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}, \quad (10)$$

where

Recall

$$= (\# \text{ of pixels correctly classified as foreground})$$

$$\times (\# \text{ of pixels classified as foreground}$$

$$\text{in the ground truth})^{-1}, \quad (11)$$

Precision

$$= \frac{\# \text{ of pixels correctly classified as foreground}}{\# \text{ of pixels classified as foreground}}.$$

For the first sequence of images, trial 1, the wave tank was set up to produce waves with a height of 0.05 meters and a period of 1.5 seconds. Based upon the manual segmentation of the desired object of interest from the sequence of images, it was calculated that the maximum expected movement of the desired object of interest per image in the sequence was

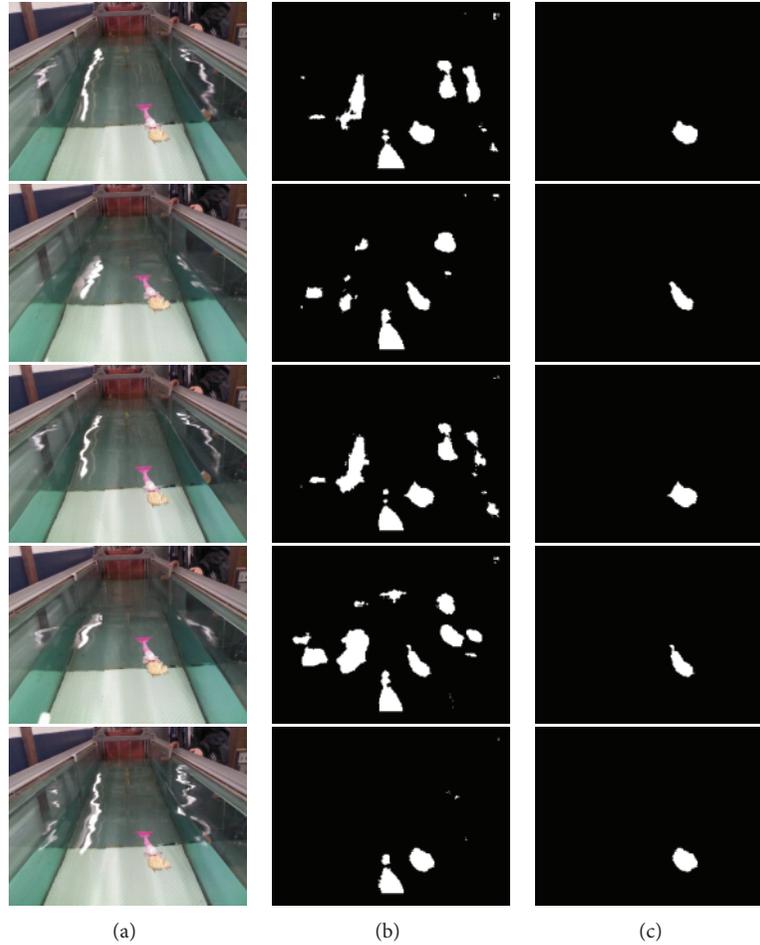


FIGURE 6: Example of five images from trial 2. (a) Original image. (b) Resulting images from fuzzy rule-based technique. (c) Resulting images from our improvement technique.

a pixel radius of 5.1. However, when the fuzzy rule-based method was performed on the sequence of images of trial 1, the method was unable to segment the desired object of interest. Based upon the fact that, of the three trials that were performed, trial 1 had the smallest movement per image, we believe that the fuzzy rule-based method failed on this image sequence due to the limited movement of the desired object of interest that caused the desired object of interest to be incorporated into the background of the image. Since the fuzzy rule-based method failed to detect the desired object of interest among the detected components, we were unable to test our improvement to the method.

For trial 2, the wave tank was set to produce waves with a height of 0.08 meters and a period of 1.5 seconds. Based upon the manual segmentation of the desired object of interest from the sequence of images from trial 2, it was calculated that the maximum expected movement of the desired object of interest per image in the sequence was a pixel radius of 6.5. Using the calculated dynamic interaction information and the results from the fuzzy rule-based method, we applied our improvement to detect and track the desired object of interest from the sequence of images. As shown in Table 1, our improvement to the fuzzy rule-based technique yielded

significantly improved results from the original fuzzy rule-based technique. Table 1 shows the recall, precision, and F -measure values for five example images from each of the two image sequences where our improved method was performed and the corresponding fuzzy rule-based technique results. Five consecutive example images and corresponding fuzzy rule-based technique images are shown in Figure 6 from the image sequence of trial 2. For trial 3, the wave tank was set to produce waves with a height of 0.1 meters and a period of 1.5 seconds. The calculated maximum expected movement per image based on the manual segmentation of the desired object of interest was a pixel radius of 6.4. Again, we applied our improvement to the fuzzy rule-based technique for the trial 3 sequence of images. Five consecutive example images and corresponding fuzzy rule-based technique images are shown in Figure 7 from the image sequence of trial 3.

7. Conclusion

In this paper, we hypothesized that the dynamic interaction between an object of interest and the background can provide

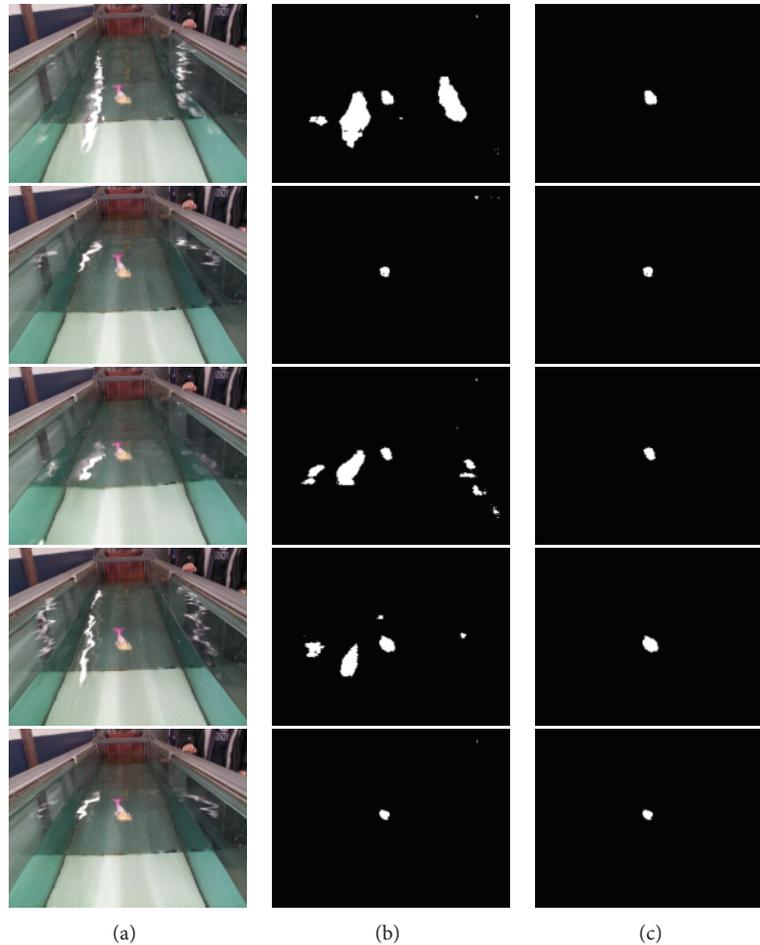


FIGURE 7: Example of five images from trial 3. (a) Original image. (b) Resulting images from fuzzy rule-based technique. (c) Resulting images from our improvement technique.

useful information and by understanding and modeling the dynamic interaction between an object of interest and background, we could improve the performance of state-of-the-art object detection techniques. After implementing two current state-of-the-art techniques and evaluating the performance when conducted on our dynamic water tank environment, we observed that the fuzzy rule-based technique proposed by Chua et al. was able to segment the frames of image sequences into a number of distinct components, one of which was the desired object of interest. Understanding and modeling the dynamic interaction between the object of interest and the background by manually segmenting and recording the movement information for a specific dynamic environment (i.e., wave height and period) allowed us to develop an algorithm to incorporate the dynamic interaction information into the segmentation process. Using our algorithm and the fuzzy rule-based technique, we performed segmentation on three image sequences from our wave tank data set. During the first trial, the fuzzy rule-based technique was unable to detect the object of interest among the detected components. Since the fuzzy

rule-based technique failed, we were unable to implement our algorithm for the first trial. However, on the second and third trial, the fuzzy rule-based technique performed adequately enough for us to implement our algorithm and compare the results of our algorithm against the results of the fuzzy rule-based technique. Based on the recall, precision, and F -measure data calculated for the trials, our proposed algorithm significantly improves upon the results observed from the fuzzy rule-based algorithm proposed by Chua et al. The experimental results achieved in this paper demonstrate convincing evidence that the dynamic interaction between an object of interest and background provides valuable information that can be utilized to improve segmentation results.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

TABLE 1: Recall, precision, and F -measure data.

Image sequence	Metric	Fuzzy rule-based method	Our method
		Trial 2	
Image 1	Recall	0.28261	0.56522
	Precision	0.023851	0.68421
	F -Measure	0.043989	0.61905
Image 2	Recall	0.31382	0.62763
	Precision	0.026282	0.91781
	F -Measure	0.048502	0.74548
Image 3	Recall	0.29924	0.59848
	Precision	0.022281	0.60305
	F -Measure	0.041473	0.60076
Image 4	Recall	0.31504	0.62053
	Precision	0.022303	0.93525
	F -Measure	0.041657	0.74605
Image 5	Recall	0.3093	0.6186
	Precision	0.025775	0.6838
	F -Measure	0.047585	0.64957
		Trial 3	
Image 1	Recall	0.76389	0.76389
	Precision	0.017909	0.67073
	F -Measure	0.034998	0.71429
Image 2	Recall	0.54	0.54
	Precision	0.016232	0.8617
	F -Measure	0.031518	0.66393
Image 3	Recall	0.62581	0.62581
	Precision	0.017382	0.776
	F -Measure	0.033824	0.69286
Image 4	Recall	0.81006	0.81006
	Precision	0.026571	0.68075
	F -Measure	0.051455	0.7398
Image 5	Recall	0.50562	0.50562
	Precision	0.018072	0.98901
	F -Measure	0.034897	0.66914

Acknowledgment

The project was in part supported by National Science Council (NSC) Programs NSC101-2221-E-008-039-MY3 and NSC-102-3113-P-007-014.

References

- [1] G. Xue, J. Sun, and L. Song, "Dynamic background subtraction based on spatial extended center-symmetric local binary pattern," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '10)*, pp. 1050–1054, July 2010.
- [2] Y. Benezeth, P.-M. Jodoin, B. Emile, and H. Laurent, "Comparative study of background subtraction algorithms," *Journal of Electronic Imaging*, vol. 19, no. 3, Article ID 033003, 2010.
- [3] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, December 2008.
- [4] S. Brutzer, B. Höferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1937–1944, June 2011.
- [5] S. Herrero and J. Bescos, "Background subtraction techniques: systematic evaluation and comparative analysis," in *Proceedings of the 11th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS '09)*, pp. 33–42, Berlin, Germany, 2009.
- [6] S. Panahi, S. Sheikhi, S. Hadadan, and N. Gheissari, "Evaluation of background subtraction methods," in *Proceedings of the Digital Image Computing: Techniques and Applications (DICTA '08)*, pp. 357–364, December 2008.
- [7] D. H. Parks and S. S. Fels, "Evaluation of background subtraction algorithms with post-processing," in *Proceedings of the IEEE 5th International Conference on Advanced Video and Signal Based Surveillance (AVSS '08)*, pp. 192–199, September 2008.
- [8] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "P finder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.
- [9] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 246–252, June 1999.
- [10] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 28–31, August 2004.
- [11] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006.
- [12] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 827–832, 2005.
- [13] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1162, 2002.
- [14] A. M. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proceedings of the 6th European Conference on Computer Vision—Part II (ECCV '00)*, pp. 751–767, London, UK, 2000.
- [15] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 1305–1312, October 2003.
- [16] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1778–1792, 2005.
- [17] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 171–177, 2010.

- [18] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, vol. 5, pp. 3061–3064, October 2004.
- [19] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.
- [20] O. Barnich and M. van Droogenbroeck, "ViBe: a universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.
- [21] O. Barnich and M. van Droogenbroeck, "ViBE: a powerful random technique to estimate the background in video sequences," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 945–948, April 2009.
- [22] Y. Li, W. Chen, and R. Jiang, "The integration adjacent frame difference of improved ViBe for foreground object detection," in *Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '11)*, pp. 1–4, September 2011.
- [23] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 293–313, 2002.
- [24] M. van Droogenbroeck and O. Paquot, "Background subtraction: experiments and improvements for ViBe," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '12)*, pp. 32–37, 2012.
- [25] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.
- [26] S. Zhang, H. Yao, and S. Liu, "Dynamic background modeling and subtraction using spatio-temporal local binary patterns," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '08)*, pp. 1556–1559, October 2008.
- [27] T. W. Chua, K. Leman, and Y. Wang, "Fuzzy rule-based system for dynamic texture and color based background subtraction," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ '12)*, pp. 1–7, 2012.
- [28] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425–436, 2009.
- [29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [30] E. M. Palmer, P. J. Kellman, and T. F. Shipley, "A theory of dynamic occluded and illusory object perception," *Journal of Experimental Psychology*, vol. 135, no. 4, pp. 513–541, 2006.

Research Article

Efficient Stereo Matching with Decoupled Dissimilarity Measure Using Successive Weighted Summation

Cheng-Tao Zhu,¹ Yau-Zen Chang,² Huai-Ming Wang,² Kai He,¹
Shih-Tseng Lee,³ and Chung-Fu Lee²

¹ School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China

² Department of Mechanical Engineering, Chang Gung University, Taoyuan 33302, Taiwan

³ Department of Neurosurgery, Chang Gung Memorial Hospital, Taoyuan 33305, Taiwan

Correspondence should be addressed to Yau-Zen Chang; zen@mail.cgu.edu.tw

Received 31 October 2013; Accepted 16 December 2013; Published 16 January 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Cheng-Tao Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Developing matching algorithms from stereo image pairs to obtain correct disparity maps for 3D reconstruction has been the focus of intensive research. A constant computational complexity algorithm to calculate dissimilarity aggregation in assessing disparity based on separable successive weighted summation (SWS) among horizontal and vertical directions was proposed but still not satisfactory. This paper presents a novel method which enables decoupled dissimilarity measure in the aggregation, further improving the accuracy and robustness of stereo correspondence. The aggregated cost is also used to refine disparities based on a local curve-fitting procedure. According to our experimental results on Middlebury benchmark evaluation, the proposed approach has comparable performance when compared with the selected state-of-the-art algorithms and has the lowest mismatch rate. Besides, the refinement procedure is shown to be capable of preserving object boundaries and depth discontinuities while smoothing out disparity maps.

1. Introduction

Stereo vision is the technique of constructing a 3D description of the scene from stereo image pairs, which is important in many computer vision tasks such as inspection [1], 3D object recognition [2], robot manipulation [3], and autonomous navigation [4]. Stereo vision systems can be active or passive. Active techniques utilize ultrasonic transducers and structured light or laser to simplify the stereo matching problem. On the other hand, passive stereo vision based only on stereo image pairs is less intrusive and typically able to provide a compact and affordable solution for range sensing.

For passive stereo vision systems, stereo matching algorithms are crucial for correct and accurate depth estimation, which find for each pixel in one image the corresponding pixel in the other image. A 2D picture of displacements between corresponding pixels of a stereo image pair is named as a disparity map [5].

Reference [6] is an intensively cited classification of stereo matching algorithms for rectified image pairs. The paper divides most of the algorithms into four sequential parts: matching cost calculation, cost aggregation, disparity computation, and disparity refinement. Among the steps, cost aggregation determines the performance of an algorithm in terms of computational complication and correctness.

Cost aggregation can be local [7–12] or global [13–16], based on differences in the range of supporting regions or windows. Global methods assume that the scene is piecewise smooth and search for disparity assignments over the whole stereo pair [6], which requires high computational operation. The local methods, also known as window-based, typically require less memory and computation. As a result, the window-based algorithms are popular for fast disparity calculations [17].

Local methods tend to be sensitive to noise; however, and its correctness at regions with sparse texture or near depth

discontinuities relies on proper selection of window size. To overcome this problem, [7] proposed variable windows for matching calculation, while [8] proposed multiple windows to enhance correctness at regions near depth discontinuities. Nevertheless, performance of these approaches is limited, since same aggregation weights are applied over the windows.

Recent years have seen adaptive support weight approaches [9] to improve quality of disparity maps. Unfortunately, these approaches require independent support weights calculation for each pixel and dramatically increase computational complexity.

To simplify computation, [10] introduced joint histogram to reduce the search region of disparity and [11] proposed the usage of a sparse Census mask. A summed normalized cross-correlation was proposed in [12] to calculate matching cost in two stages. Segmentation and plane fitting on disparity planes [13–16] are also popular to improve accuracy of disparity, but the performance relies on correctness in both segmentation and plane fitting.

An effective local stereo matching algorithm is introduced in [18], which significantly simplifies the intensity-dependent aggregation procedure of local methods. The algorithm aggregates cost values effectively in terms of bilateral filtering by only four passes along regions, called separable successive weighted summation (SWS), eliminating iteration and support area dependency. However, the dissimilarity measures are coupled, which significantly restrict the flexibility in weighting the aggregated costs.

In this paper, we present an improved stereo matching algorithm. Similar to [18], our algorithm uses whole regions as matching primitives to assess disparity based on SWS among horizontal and vertical directions. We also use the basic metrics, such as truncated sum of absolute difference and truncated absolute gradient difference, as dissimilarity measure to provide a trade-off between accuracy and complexity.

The main contribution of this paper is to afford a decoupled aggregation algorithm to access the stereo matching cost under the framework of SWS. The algorithm is simply yet efficient as well as robust. In addition, the resultant disparity map is in a discrete space, which is unsuitable for image-based rendering. We propose a subpixel refinement technique that employs inferior candidate disparities, rather than spatial neighbors, to smooth out discrete values in the disparity map. By this arrangement for curve-fitting, even regions near discontinuous depth can be correctly refined. Moreover, this technique increases the resolution of a stereo algorithm with marginal additional computation.

2. Aggregation Algorithm Design

Our stereo matching algorithm consists of three main stages. First, initial cost values are calculated based on dissimilarity measure between pixels in the reference and target images, and the costs are aggregated using the proposed method. Second, we perform initial disparity estimation by the use of a winner-takes-all minimum search based on the aggregated costs. Third, we check differences between the disparity values of corresponding pixel pairs for the existence of

obscured regions and patch them by the smallest disparity values of nearby regions. Finally, the disparity map is refined by a proposed curve-fitting procedure.

2.1. Cost Definition. Assuming that the image pair is rectified and horizontally aligned, two dissimilarity measures between the pixels on the reference image and the target image are used in this work.

The truncated absolute difference cost, e_D , is defined as the absolute difference between the intensity of pixels with the corresponding pixel being shifted d pixels, along horizontal direction on the reference image:

$$e_D(x, y, d) = \min \left[\sum_{n=1}^3 |I_R(x, y, n) - I_T(x + d, y, n)|, h_D \right], \quad (1)$$

where I_R and I_T are the intensities of the pixels on reference and target images with $n = 1, 2, 3$ corresponding to three-channel colors: R, G, and B, respectively, (x, y) are pixel coordinates, and h_D is a threshold value for e_D with $h_D > 0$. The use of a threshold to restrict cost value has been a well-adopted practice to reduce the effects of noise and potential mismatch in obscured regions.

Besides, the truncated absolute gradient difference cost is defined as

$$e_G(x, y, d) = \min \left[\sum_{n=1}^3 |\nabla_x I_R(x, y, n) - \nabla_x I_T(x + d, y, n)| + \sum_{n=1}^3 |\nabla_y I_R(x, y, n) - \nabla_y I_T(x + d, y, n)|, h_G \right], \quad (2)$$

where ∇_x and ∇_y are the horizontal and vertical gradient operators, respectively, and h_G is a threshold value for e_G with $h_G > 0$.

2.2. Cost Aggregation. Aggregation of primary costs determines correctness and accuracy of disparity estimation [19–21]. Based on the observation that costs, such as the absolute difference and the absolute gradient difference, represent different dissimilarity characteristics, they should be independently weighted in aggregation. This section proposes a new method that is compatible with separable successive weighted summation (SWS) [18] while efficiently provides decoupled dissimilarity aggregation for robust stereo matching.

Once the e_D and e_G cost measures are obtained, as was defined in the last subsection, the aggregated cost function \bar{C} is set as the weighted sum of both measures according to a weighting factor, λ :

$$\begin{aligned} \bar{C}(x, y, d) &= \lambda \sum_{i=1}^m \sum_{j=1}^n W_{ix} W_{jy} e_D(i, j, d) \\ &\quad + (1 - \lambda) \sum_{i=1}^m \sum_{j=1}^n U_{ix} U_{jy} e_G(i, j, d) \\ &\triangleq \lambda \widehat{C}_D(x, y, d) + (1 - \lambda) \widehat{C}_G(x, y, d). \end{aligned} \quad (3)$$

In (3), definition of the weightings, W_{ix} , W_{jy} , U_{ix} , and U_{jy} , are based on the operational principles of bilateral filters [22] to dramatically reduce computation:

$$W_{ix} = \begin{cases} \prod_{k=i+1}^x \exp\left(\frac{\sqrt{\sum_{n=1}^3 [I_R(k, y, n) - I_T(k-1, y, n)]^2}}{-2\alpha}\right), & i < x \\ 1, & i = x \\ \prod_{k=x+1}^i \exp\left(\frac{\sqrt{\sum_{n=1}^3 [I_R(k, y, n) - I_T(k-1, y, n)]^2}}{-2\alpha}\right), & i > x \end{cases} \quad (4)$$

$$W_{jy} = \begin{cases} \prod_{k=j+1}^y \exp\left(\frac{\sqrt{\sum_{n=1}^3 [I_R(x, k, n) - I_T(x, k-1, n)]^2}}{-2\alpha}\right), & j < y \\ 1, & j = y \\ \prod_{k=y+1}^j \exp\left(\frac{\sqrt{\sum_{n=1}^3 [I_R(x, k, n) - I_T(x, k-1, n)]^2}}{-2\alpha}\right), & j > y \end{cases} \quad (5)$$

$$U_{ix} = \begin{cases} \prod_{k=i+1}^x \exp\left(\frac{\sqrt{\sum_{n=1}^3 [\nabla_x I_R(k, y, n) - \nabla_x I_T(k-1, y, n)]^2}}{-2\beta}\right), & i < x \\ 1, & i = x \\ \prod_{k=x+1}^i \exp\left(\frac{\sqrt{\sum_{n=1}^3 [\nabla_x I_R(k, y, n) - \nabla_x I_T(k-1, y, n)]^2}}{-2\beta}\right), & i > x \end{cases} \quad (6)$$

$$U_{jy} = \begin{cases} \prod_{k=j+1}^y \exp\left(\frac{\sqrt{\sum_{n=1}^3 [\nabla_y I_R(x, k, n) - \nabla_y I_T(x, k-1, n)]^2}}{-2\beta}\right), & j < y \\ 1, & j = y \\ \prod_{k=y+1}^j \exp\left(\frac{\sqrt{\sum_{n=1}^3 [\nabla_y I_R(x, k, n) - \nabla_y I_T(x, k-1, n)]^2}}{-2\beta}\right), & j > y. \end{cases} \quad (7)$$

The values of W_{ix} and W_{jy} increase with smaller e_D , while the values of U_{ix} and U_{jy} rise with lesser e_G . Also, the weightings are the multiples of horizontal and vertical weightings, which decrease with the increase of distances to the reference pixels. Hence, for each pixel, neighboring pixels with similar intensity have higher support during the aggregation.

The aggregation is a two-dimensional convolution. To reduce the computational complexity, each convolution is further decomposed into four one-dimensional convolutions [18]. These one-dimensional convolutions operate from left to right, from right to left, from top to bottom, and from bottom to top, respectively.

Let us take the absolute difference part of \bar{C} , denoted as \bar{C}_D in (3), as an example. If we define the left-to-right weighted sum, C_D^{L2R} , as

$$C_D^{L2R}(x, j, d) \triangleq \sum_{i=1}^{x-1} W_{ix} e_D(i, j, d) + e_D(x, j, d), \quad (8)$$

from the definition of W_{ix} , (4):

$$W_{ix} = W_{i(x-1)} W_{(x-1)x}, \quad (9)$$

we have that C_D^{L2R} can be written in a recursive form

$$\begin{aligned} C_D^{L2R}(x, j, d) &= \sum_{i=1}^{x-2} W_{ix} e_D(i, j, d) + W_{(x-1)x} e_D(x-1, j, d) \\ &\quad + e_D(x, j, d) \\ &= W_{(x-1)x} \left[\sum_{i=1}^{x-2} W_{i(x-1)} e_D(i, j, d) \right. \\ &\quad \left. + e_D(x-1, j, d) \right] + e_D(x, j, d) \\ &= W_{(x-1)x} C_D^{L2R}(x-1, j, d) + e_D(x, j, d). \end{aligned} \quad (10)$$

Similarly, we may define the right-to-left weighted sum, C_D^{R2L} , as

$$C_D^{R2L}(x, j, d) \triangleq \sum_{i=x+1}^m W_{ix} e_D(i, j, d) + e_D(x, j, d), \quad (11)$$

which can also be written in a recursive form

$$C_D^{R2L}(x, j, d) = W_{x(x+1)} C_D^{R2L}(x+1, j, d) + e_D(x, j, d). \quad (12)$$

We have that

$$\begin{aligned}
\widehat{C}_D(x, y, d) &= \sum_{j=1}^n \sum_{i=1}^m W_{jy} W_{ix} e_D(i, j, d) \\
&= \sum_{j=1}^n W_{jy} \left[\sum_{i=1}^m W_{ix} e_D(i, j, d) \right] \\
&= \sum_{j=1}^n W_{jy} \left[\sum_{i=1}^{x-1} W_{ix} e_D(i, j, d) + e_D(x, j, d) \right. \\
&\quad \left. + \sum_{i=x+1}^m W_{ix} e_D(i, j, d) \right] \\
&= \sum_{j=1}^n W_{jy} \left[C_D^{L2R}(x, j, d) + C_D^{R2L}(x, j, d) \right. \\
&\quad \left. - e_D(x, j, d) \right] \\
&\triangleq \sum_{j=1}^n W_{jy} C_D^H(x, j, d).
\end{aligned} \tag{13}$$

Note that, in (13), we have defined

$$C_D^H(x, j, d) \triangleq C_D^{L2R}(x, j, d) + C_D^{R2L}(x, j, d) - e_D(x, j, d) \tag{14}$$

to simplify the following derivation.

In the vertical direction, if we define the top-to-bottom and bottom-to-top weighted sums, C_D^{T2B} and C_D^{B2T} , as

$$C_D^{T2B}(x, y, d) \triangleq \sum_{j=1}^{y-1} W_{jy} C_D^H(x, j, d) + e_D(x, y, d) \tag{15}$$

$$C_D^{B2T}(x, y, d) \triangleq \sum_{j=y+1}^n W_{jy} C_D^H(x, j, d) + e_D(x, y, d),$$

we have that

$$\widehat{C}_D(x, y, d) = C_D^{T2B}(x, y, d) + C_D^{B2T}(x, y, d) - e_D(x, y, d). \tag{16}$$

In calculating (16), both C_D^{T2B} and C_D^{B2T} are recursively obtained as

$$\begin{aligned}
C_D^{T2B}(x, y, d) &= W_{(y-1)y} C_D^{T2B}(x, y-1, d) + e_D(x, y, d), \\
C_D^{B2T}(x, y, d) &= W_{y(y+1)} C_D^{B2T}(x, y+1, d) + e_D(x, y, d).
\end{aligned} \tag{17}$$

Hence, the first part of the aggregated cost, \widehat{C}_D , can be efficiently calculated by (10), (12), and (17).

With similar procedure, we have that

$$\widehat{C}_G(x, y, d) = C_G^{T2B}(x, y, d) + C_G^{B2T}(x, y, d) - e_G(x, y, d), \tag{18}$$

where

$$C_G^{T2B}(x, y, d) \triangleq \sum_{j=1}^{y-1} U_{jy} C_G^H(x, j, d) + e_G(x, y, d), \tag{19}$$

$$C_G^{B2T}(x, y, d) \triangleq \sum_{j=y+1}^n U_{jy} C_G^H(x, j, d) + e_G(x, y, d)$$

with

$$C_G^H(x, j, d) = C_G^{L2R}(x, j, d) + C_G^{R2L}(x, j, d) - e_G(x, j, d),$$

$$C_D^{L2R}(x, j, d) \triangleq \sum_{i=1}^{x-1} W_{ix} e_D(i, j, d) + e_D(x, j, d),$$

$$C_D^{R2L}(x, j, d) \triangleq \sum_{i=x+1}^m W_{ix} e_D(i, j, d) + e_D(x, j, d). \tag{20}$$

These terms can all be written in the following recursive form:

$$C_G^{T2B}(x, y, d) = U_{(y-1)y} C_G^{T2B}(x, y-1, d) + e_G(x, y, d),$$

$$C_G^{B2T}(x, y, d) = U_{y(y+1)} C_G^{B2T}(x, y+1, d) + e_G(x, y, d),$$

$$C_G^{L2R}(x, j, d) = U_{(x-1)x} C_G^{L2R}(x-1, j, d) + e_G(x, j, d),$$

$$C_G^{R2L}(x, j, d) = U_{x(x+1)} C_G^{R2L}(x+1, j, d) + e_G(x, j, d). \tag{21}$$

2.3. Disparity Computation. In the last section, the matching cost is aggregated through weighted summation over the entire image, and the disparity which provides minimum cost is assigned to the corresponding pixel. That is, the assigned disparity for a pixel in the reference image, D_{ref} , is the one with the minimum aggregated matching cost:

$$\bar{d}_{\text{Ref}}(x, y) = \arg \min_{d \in [0, 1, \dots, d_{\text{max}}]} \bar{C}(x, y, d), \tag{22}$$

where $[0, 1, \dots, d_{\text{max}}]$ is the disparity search space with d_{max} being the maximum possible disparity value.

The initial disparity map normally contains obscured outlier regions. The disparities at these regions are significantly different when they are found from different reference images. If $\bar{d}_{\text{Ref}}^{\text{Left}}$ and $\bar{d}_{\text{Ref}}^{\text{Right}}$ are the disparities with the left and right images as reference images, respectively, we can apply the Left-Right Consistency Check (LRC) [23] to determine if a pixel is located at an obscured region:

$$\begin{aligned}
\left| \bar{d}_{\text{Ref}}^{\text{Left}}(x, y) - \bar{d}_{\text{Ref}}^{\text{Right}}\left(x - \bar{d}_{\text{Ref}}^{\text{Left}}(x, y), y\right) \right| &> 1, \\
\left| \bar{d}_{\text{Ref}}^{\text{Right}}(x, y) - \bar{d}_{\text{Ref}}^{\text{Left}}\left(x + \bar{d}_{\text{Ref}}^{\text{Right}}(x, y), y\right) \right| &> 1.
\end{aligned} \tag{23}$$

Once the obscured regions are found, occlusion handling [24] can be applied to patch them with the smallest disparity values of nearby regions, and the corresponding costs are also assigned for disparity refinement, as to be used in the next subsection.

2.4. Disparity Refinement. The disparity map obtained by the method proposed in the last section is discrete, since the disparity search space is an integer set. We propose a smoothing technique for the disparity refinement in this subsection.

Considering that the initial disparity has the smallest aggregation cost in the potential solution space, we may interpolate for refined value by fitting data sets to upward curves. Besides, rather than directly using the neighboring disparity for refinement, we use both the costs and disparity in the curve-fitting.

Assuming that $\bar{C}(x, y, \bar{d})$ is the cost value corresponding to disparity \bar{d} at (x, y) on the reference image, we denote $\bar{C}_{\bar{d}} \triangleq \bar{C}(x, y, \bar{d})$, such that $\bar{C}_{\bar{d}+1} \triangleq \bar{C}(x, y, \bar{d} + 1)$ and $\bar{C}_{\bar{d}-1} \triangleq \bar{C}(x, y, \bar{d} - 1)$, to simplify the following presentation.

Firstly, the disparity-cost sets around the pixel, $(\bar{d} - 1, \bar{C}_{\bar{d}-1})$, $(\bar{d}, \bar{C}_{\bar{d}})$, and $(\bar{d} + 1, \bar{C}_{\bar{d}+1})$, are fitted to a hyperbolic function

$$\varphi = f(\xi) = a_1\xi + \frac{a_2}{\xi} + a_3, \text{ where} \quad (24)$$

$$a_1, a_2, a_3 \in \mathbb{R}, \quad a_1 > 0, \quad a_2 > 0.$$

As the minimum value of the curve is located at $\bar{d}_1^* = \sqrt{a_2/a_1}$, we have that

$$\bar{d}_1^* = \sqrt{\frac{(\bar{d}^3 - \bar{d})(\bar{C}_{\bar{d}+1} + \bar{C}_{\bar{d}-1} - 2\bar{C}_{\bar{d}})}{\bar{d}(\bar{C}_{\bar{d}+1} + \bar{C}_{\bar{d}-1} - 2\bar{C}_{\bar{d}}) + \bar{C}_{\bar{d}+1} - \bar{C}_{\bar{d}-1}}}. \quad (25)$$

Secondly, an upward parabola equation

$$\theta = g(\zeta) = b_1\zeta^2 + b_2\zeta + b_3, \text{ where} \quad (26)$$

$$b_1, b_2, b_3 \in \mathbb{R}, \quad b_1 > 0, \quad b_2 < 0$$

is used to fit other disparity-cost sets around the pixel, $(\bar{d} - 2, \bar{C}_{\bar{d}-2})$, $(\bar{d}, \bar{C}_{\bar{d}})$, and $(\bar{d} + 2, \bar{C}_{\bar{d}+2})$. As the minimum value of the parabola is located at $\bar{d}_2^* = -b_2/2b_1$, we have

$$\bar{d}_2^* = \bar{d} - \frac{\bar{C}_{\bar{d}+2} - \bar{C}_{\bar{d}-2}}{\bar{C}_{\bar{d}+2} + \bar{C}_{\bar{d}-2} - 2\bar{C}_{\bar{d}}}. \quad (27)$$

The averaged value, $\bar{d}^* = (\bar{d}_1^* + \bar{d}_2^*)/2$, is then used as the refined disparity.

3. Experimental Study

In contrast to the approach of [18], the proposed algorithm applied independent weights to the mismatch measures while

TABLE 1: Comparison of percentage of miss-matches among the proposed algorithm and [18] in nonoccluded regions.

Image set	Algorithm	
	Proposed	InfoPerm [18]
Tsukuba	3.53%	5.99%
Venus	0.34%	1.32%
Teddy	6.09%	6.20%
Cones	3.60%	4.58%

preserving comparable computational efficiency. Below, we rewrite the aggregated cost function, (3), for comparison:

$$\begin{aligned} \bar{C}(x, y, d) = & \lambda \sum_{i=1}^m \sum_{j=1}^n W_{ix} W_{jy} e_D(i, j, d) \\ & + (1 - \lambda) \sum_{i=1}^m \sum_{j=1}^n U_{ix} U_{jy} e_G(i, j, d). \end{aligned} \quad (28)$$

The aggregated cost function of [18] is equivalent to

$$\begin{aligned} C'(x, y, d) \\ = & \sum_{i=1}^m \sum_{j=1}^n \bar{W}_{ix} \bar{W}_{jy} [\mu e_D(i, j, d) + (1 - \mu) e_{\text{CENSUS}}(i, j, d)], \end{aligned} \quad (29)$$

where e_{CENSUS} is the Census measure [25, 26]. It is clear from the comparison between (28) and (29) that the proposed algorithm (28) enables separate weightings on different measures in the aggregation.

A performance comparison between the proposed method and the algorithm of [18], denoted as InfoPerm [18], using the Middlebury stereo test bench [27, 28] is presented in Figure 1. In the computation, the parameters of the proposed algorithm, h_D , h_G , α , β , and λ , are selected as 22, 38, 32, 23, and 0.6, respectively.

In Figure 1, and hence the following presentations, disparities with errors larger than 0.5 disparity levels are named as mismatches, which are denoted in gray at non-occluded regions and black for occluded regions. The percentages of mismatches are further calculated and summarized in Table 1, which shows that the proposed algorithm outperforms InfoPerm [18] in correctness at non-occluded regions in the benchmark tests.

In addition to InfoPerm [18], several state-of-the-art methods, such as SNCC [12], HistAggr2 [10], RTCensus [11], InfoPerm [18], AdaptWeight [9], FeatureGC [13], ObjectStereo [15], and AdaptingBP [16] were also implemented on the Middlebury stereo test bench [25, 26] for a complete performance comparison. Among them, SNCC [12], HistAggr2 [10], RTCensus [11], InfoPerm [18], and AdaptWeight [9] are local stereo match algorithms, while FeatureGC [13], ObjectStereo [15], and AdaptingBP [16] are global techniques.

Figure 2 shows a comparison of the disparity maps between RTCensus [11], SNCC [12], AdaptWeight [9] and the proposed approach, where all the disparity maps have

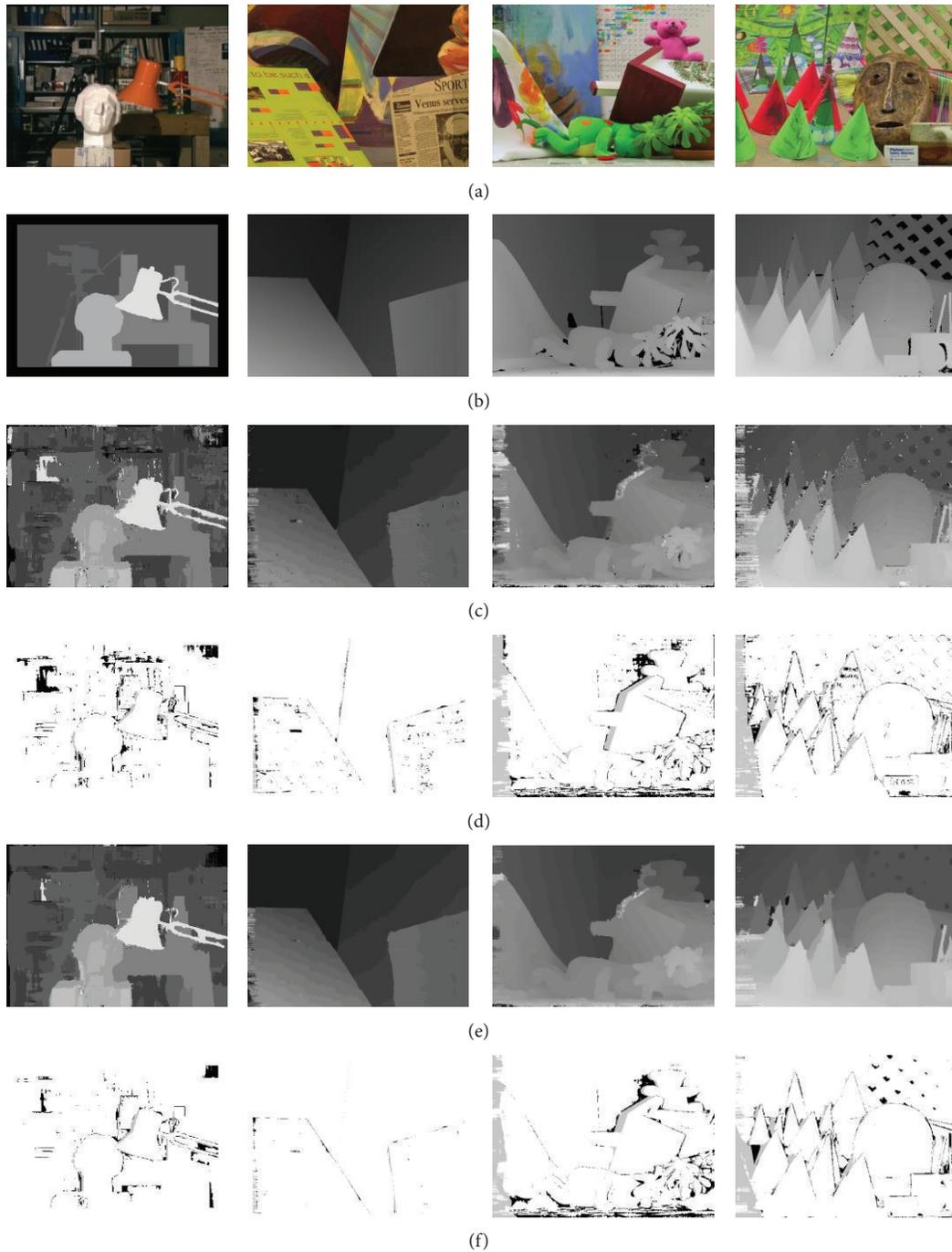


FIGURE 1: A comparison of disparity estimation performance between [18] and the proposed algorithms. Top-to-bottom: (1) first row: images (Tsukuba, Venus, Teddy, and Cones) from Middlebury stereo database, (2) second row: ground truth disparity maps, (3) third row: disparity maps by the InfoPerm [18] algorithm, (4) fourth row: mismatches of the InfoPerm [18] algorithm, (5) fifth row: disparity maps by the proposed algorithm, and (6) sixth row: mismatches of the proposed algorithm.

been refined. The results show that the proposed method has comparable performance with these state-of-the-art methods, and the refinement strategy, introduced in Section 2.4, is able to preserve clear boundaries.

The complete comparison of the mismatch rates between these algorithms is summarized in Table 2. In the table,

“nonocc.” denotes the pixels in the non-occluded region, and “disc.” represents the discontinuous but visible pixels near the occluded regions. According to Table 2, the proposed algorithm outperforms AdaptWeight [9] in all of the mismatch evaluations, although it seems inferior to AdaptWeight [9] in generating sharp boundary, as shown in Figure 1.



FIGURE 2: Top-to-bottom: (1) color views (Tsukuba, Venus, Teddy, and Cone), (2) ground truth disparity maps, (3) disparity maps via RTCensus [11], (4) disparity maps via SNCC [12], (5) disparity maps via AdaptWeight [9], and (6) disparity maps via the proposed approach.

It is also interesting to note that the local stereo matching algorithms, such as RTCensus [11] and SNCC [12], outperform other algorithms for the Teddy and Cones image pairs. Nevertheless, the global stereo matching algorithms, such as FeatureGC [13] and AdaptingBP [16], perform better for the Tsukuba and Venus image pairs. This observation indicates that both the local and global approaches are case-sensitive. However, the proposed approach has comparable

performance in all of the cases and has the lowest mismatch rate in the benchmark evaluation.

4. Conclusion

Stereo matching algorithms are crucial for correct and accurate depth estimation in passive stereo vision systems. A stereo matching algorithm processes rectified stereo image

TABLE 2: Comparison of mismatch percentage among the proposed algorithm and several representative algorithms.

Algorithms	Tsukuba			Venus			Teddy			Cones			Average percent of mismatch
	nonocc.	all	disc.										
SNCC [12]	11.3	12.3	27.5	2.35	3.23	15.4	10.6	15.2	28.6	4.71	11.1	13.2	13.0
HistAggr2 [10]	15.2	15.7	16.4	6.44	7.00	12.5	11.5	18.2	24.4	7.90	13.7	14.2	13.6
RTCensus [11]	12.9	14.1	28.1	3.67	4.63	17.8	11.4	18.6	27.7	5.54	11.8	15.9	14.4
InfoPerm [18]	25.7	26.2	21.2	8.64	9.34	15.0	15.0	22.1	29.2	7.68	15.1	15.1	17.5
AdaptWeight [9]	18.1	18.8	18.6	7.77	8.40	15.8	17.6	23.9	34.0	14.0	19.7	20.6	18.1
FeatureGC [13]	8.22	8.86	13.3	4.58	4.73	10.1	14.7	17.0	32.5	11.5	18.0	21.0	13.7
ObjectStereo [15]	16.4	16.8	16.1	2.56	2.67	7.69	19.6	22.7	30.3	16.3	20.7	19.7	16.0
AdaptingBP [16]	19.1	19.3	17.4	4.84	5.08	7.84	12.8	16.7	26.3	7.02	13.2	14.0	13.6
Proposed	9.12	10.2	15.0	1.04	1.72	8.35	11.2	17.0	26.8	6.36	12.6	15.1	11.2

nonocc.: the pixels in the nonoccluded region; disc.: the visible pixels near the discontinuous regions.

pairs to generate the disparity map, which is used to calculate the depth image (z-map), and hence the 3D point cloud in camera coordinates. For practical applications, the algorithms should require less computational resources and provide precise disparity maps.

In this paper, we proposed an efficient stereo matching algorithm and a refinement strategy for the disparity maps. The algorithm effectively aggregates cost values in terms of bilateral filtering by only four passes along regions, which is able to provide a decoupled dissimilarity measure aggregation while preserving computational efficiency. Besides, the refinement strategy is a simple application of the aggregated costs that use both the costs and disparity in the curve-fitting, rather than directly using the neighboring disparity for refinement.

Experimental results using the Middlebury stereo test bench [25, 26] show that the algorithm has comparable performance with the state-of-the-art algorithms and outperforms the representative algorithms in the overall mismatch rate.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper was sponsored by Chang Gung Memorial Hospital, Chang Gung University, and the National Science Council, Taiwan, under Contract nos. CMRPD1C0021, CMRPD2C0051, NSC 100-2221-E-182-008, NSC 101-2221-E-182-006, and NSC 102-2221-E-182-073 and the National Science Foundation of China, under Contract no. 61271326.

References

- [1] Á. González, M. Á. Garrido, D. F. Llorca et al., "Automatic traffic signs and panels inspection system using computer vision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 485–499, 2011.
- [2] F. Tombari, F. Gori, and L. di Stefano, "Evaluation of stereo algorithms for 3D object recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops '11)*, pp. 990–997, Barcelona, Spain, November 2011.
- [3] K. Ohno, K. Kense, E. Takeuchi, and Z. Lei, "Unknown object modeling on the basis of vision and pushing manipulation," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '11)*, pp. 1942–1948, Phuket, Thailand, December 2011.
- [4] K. Schauwecker and A. Zell, "On-board dual-stereo-vision for autonomous quadrotor navigation," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS '13)*, pp. 333–342, Atlanta, Ga, USA, May 2011.
- [5] V. Murino, U. Castellani, and A. Fusiello, "Disparity map restoration by integration of confidence in Markov random fields models," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '01)*, pp. 29–32, Thessaloniki, Greece, October 2001.
- [6] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1–3, pp. 7–42, 2002.
- [7] Y. Boykov, O. Veksler, and R. I. Zabí, "A variable window approach to early vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1283–1294, 1998.
- [8] H. Hirschmüller, P. R. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *International Journal of Computer Vision*, vol. 47, no. 1–3, pp. 229–246, 2002.
- [9] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 650–656, 2006.
- [10] D. Min, J. Lu, and M. N. Do, "Joint histogram based cost aggregation for stereo matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2539–2545, 2013.
- [11] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze, "A fast stereo matching algorithm suitable for embedded real-time systems," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1180–1202, 2010.
- [12] N. Einecke and J. Eggert, "A two-stage correlation method for stereoscopic depth estimation," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA '10)*, pp. 227–234, Sydney, Australia, December 2010.

- [13] G. Saygili, L. van der Maaten, and E. A. Hendriks, "Feature-based stereo matching using graph cuts," in *Proceedings of ASCI-IPA-SIKS Tracks, IPA Fall Days/ICT.OPEN*, Veldhoven, The Netherlands, November 2011.
- [14] Z.-F. Wang and Z.-G. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [15] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha, "Object stereo Joint stereo matching and object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3081–3088, Washington, DC, USA, June 2011.
- [16] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 15–18, Hong Kong, August 2006.
- [17] R. Yang and M. Pollefeys, "Multi-resolution real-time stereo on commodity graphics hardware," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. 1/211–1/217, Washington, DC, USA, June 2003.
- [18] C. Cigla and A. A. Alatan, "Efficient edge-preserving stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVWorkshops '11)*, pp. 696–699, Barcelona, Spain, November 2011.
- [19] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2012.
- [20] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Computer Vision—ECCV 2010*, vol. 6311 of *Lecture Notes in Computer Science*, pp. 1–14, Springer, Heidelberg, Greece, 2010.
- [21] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza, "Linear stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1708–1715, Barcelona, Spain, November 2011.
- [22] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the 6th IEEE International Conference on Computer Vision (ICCV '98)*, pp. 839–846, Bombay, India, January 1998.
- [23] P. Fua, "Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities," in *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI '91)*, vol. 2, pp. 1292–1298, Sydney, Australia, 1991.
- [24] L. D. Stefano, M. Marchionni, and S. Mattoccia, "A fast area-based stereo matching algorithm," *Image and Vision Computing*, vol. 22, no. 12, pp. 983–1005, 2004.
- [25] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Computer Vision—ECCV 1994*, *Lecture Notes in Computer Science*, pp. 151–158, Springer, 1994.
- [26] N. Y.-C. Chang, T.-H. Tsai, B.-H. Hsu, Y.-C. Chen, and T.-S. Chang, "Algorithm and architecture of disparity estimation with mini-census adaptive support weight," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 792–805, 2010.
- [27] "Middlebury stereo vision database," <http://vision.middlebury.edu/stereo/>.
- [28] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.

Research Article

Poissonian Image Deconvolution via Sparse and Redundant Representations and Framelet Regularization

Yu Shi, Houzhang Fang, and Guoyou Wang

Science and Technology on Multi-Spectral Information Processing Laboratory, Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China

Correspondence should be addressed to Guoyou Wang; gywang@mail.hust.edu.cn

Received 24 September 2013; Accepted 6 December 2013; Published 16 January 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Yu Shi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Poissonian image deconvolution is a key issue in various applications, such as astronomical imaging, medical imaging, and electronic microscope imaging. A large amount of literature on this subject is analysis-based methods. These methods assign various forward measurements of the image. Meanwhile, synthesis-based methods are another well-known class of methods. These methods seek a reconstruction of the image. In this paper, we propose an approach that combines analysis with synthesis methods. The method is proposed to address Poissonian image deconvolution problem by minimizing the energy functional, which is composed of a sparse representation prior over a learned dictionary, the data fidelity term, and framelet based analysis prior constraint as the regularization term. The minimization problem can be efficiently solved by the split Bregman technique. Experiments demonstrate that our approach achieves better results than many state-of-the-art methods, in terms of both restoration accuracy and visual perception.

1. Introduction

Poissonian image deconvolution appears in various applications such as astronomical imaging [1], medical imaging [2], and electronic microscope imaging [3]. It aims to reconstruct a high quality image u from the degraded image g . Mathematically, the process of Poissonian image deconvolution can be generally modeled by

$$g = \mathcal{P}(Hu), \quad (1)$$

where $H \in \mathbb{R}^{N \times N}$ denotes the point spread function (PSF), $u \in \mathbb{R}^N$ denotes the unknown image to be estimated, \mathcal{P} denotes the Poisson noise process, $g \in \mathbb{R}^N$ denotes the blurred noisy image, and N is the length of image vector which is stacked by columns.

It is known that Poissonian image deconvolution is a typical ill-posed inverse problem. In general, the solution of (1) is not unique. Prior knowledge of image, including analysis-based and synthesis-based priors, can be used to address this problem. For an overview of the two classes of

priors, we refer to [4]. Analysis-based priors are frequently used as regularization term in energy functional $E(u)$

$$E(u) = \inf_u \int_{\Omega} Hu - g \log Hu \, dx + R(u), \quad (2)$$

where $R(u)$ is the regularization constraint term. Two main analysis-based methods have been proposed to solve problem (2): the total variation (TV) [5, 6] based methods and wavelet frames-based methods. TV-based methods have shown good performance on blurred images for discontinuous solution and edge-preserving advantages. Many authors have combined the TV regularization term with variant methods to solve problem (2). For example, Sawatzky et al. combined expectation-maximization algorithm with TV regularization [7] in positron emission tomography. Setzer et al. considered using TV regularization term with split Bregman techniques [8]. TV-based methods can remove noise in flat region effectively. However, they lose details in texture region. Furthermore, the inherent problem of TV-based methods is that they favor a piecewise constant solution, which causes staircase effects in flat region.

To overcome disadvantages of TV-based methods, several authors are interested in frame-based methods that can be regarded as analysis regularization methods as well. It is assumed that majority of nature images have sparse approximation under some redundant tight frame systems including wavelet transform, Gabor transform, curvelets, and framelets [9]. Chen and Cheng considered a hybrid variational model [10] that describes a cartoon part by TV and a detailed part by the sparse representation over the wavelet basis. Figueiredo and Bioucas-Dias proposed an alternating direction method [11] using frame-based analysis regularizer. Fang et al. presented blind Poissonian images deconvolution with framelet regularization [12]. He considered framelet-based analysis prior and combination of framelet and total variation analysis priors in [13] as well.

Dictionary learning [14–20] based sparse representation has been extensively studied, and significant progress has been made in the development of modern dictionaries learning for synthesis-based sparse representation of image. Natural images contain repeated patterns such as texture region, flat region, and smooth region. Hence, they can be described as linear combinations of a few atoms from a dictionary. Dictionary learning for synthesis model in image denoising was proposed by Elad and Aharon. They presented the method via sparse and redundant representations over learned dictionaries, called K-SVD [16]. This method is performed to remove the additive noise. Huang et al. attempted to use the K-SVD method [17] for multiplicative noise removal which followed. It was proven that K-SVD method could perform well just by transforming the multiplicative noise to additive noise in logarithmic domain. Similarly, Poisson noise can be transformed to additive Gaussian noise with unitary variance by Anscombe root transformation (ART) [18] and then removed by K-SVD method [15]. In fact, the variance stabilization often has been questioned for the poor numerical results. Therefore, the denoising algorithm should be specifically designed for Poisson noise. Ma et al. proposed dictionary learning for Poisson noise removal [21].

We still have work to do, on dictionary learning for Poisson noise removal, because TV regularizer and dictionary learning (refer to [21]) do not match well. TV regularizer can reduce artifacts in smooth regions caused by patch-based priors of dictionary learning; however, it smooths the fine details in texture region. Motivated by analysis and synthesis based methods, we combine analysis with synthesis method for Poissonian image deconvolution. We choose B-spline framelet as the analysis regularization term, because framelet transform has the ability of multiple-resolution analysis. Different framelet masks reflect different orders of difference operators, which can adaptively capture multiscale edge structures in images. Besides, synthesis-based method is integrated into the proposed method using a sparse representation prior over a learned dictionary. At last, split Bergman method is used to address optimization problem. Bergman variables make the proposed method stable and regularization parameters do not need to be updated in Bergman iteration process.

The rest of this paper is organized as follows. The K-SVD algorithm and B-spline framelet are briefly reviewed

in Section 2. The Poissonian image deconvolution model is proposed in Section 3. Numerical experiments are given to demonstrate the qualities of the restored images in Section 4. Conclusions are given in Section 5.

2. Review of Prior Works

In this section, we review K-SVD algorithm and B-spline framelet briefly. K-SVD algorithm is a classical sparse representation method, and it will be used in our proposed model. Framelet draws many attentions by its multiscale characteristic.

2.1. K-SVD Algorithm. We consider the image patch x of size $\sqrt{n} \times \sqrt{n}$ and define dictionary $D \in \mathbb{R}^{n \times k}$, where $k > n$. For each image patch, x can be represented sparsely over the dictionary; that is, the solution of

$$\hat{\alpha} = \arg \min_{\alpha} \|\alpha\|_0 \quad \text{subject to } D\alpha \approx x \quad (3)$$

is very sparse. The notation $\|\alpha\|_0$ stands for the count of the nonzero entries in α . Considering the image patch x contaminated by an additive zero-mean white Gaussian noise with standard deviation δ , the MAP estimator for denoising this image patch is built by solving

$$\hat{\alpha} = \arg \min_{\alpha} \|D\alpha - x\|_2^2 + \mu \|\alpha\|_0. \quad (4)$$

If each image patch x of image X can be described by (4), the MAP estimator of image X will be described by

$$\begin{aligned} \hat{\alpha}_{ij}, \hat{X} = \arg \min_{\alpha_{ij}, X} \lambda \|X - Y\|_2^2 + \sum_{ij} \|D\alpha_{ij} - R_{ij}X\|_2^2 \\ + \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_0, \end{aligned} \quad (5)$$

where Y denotes the degraded image and each image patch $x_{ij} = R_{ij}X$, R_{ij} is an $n \times N$ matrix that extracts the (ij) block from the image X . Instead of addressing both unknowns $\hat{\alpha}_{ij}$ and X together, a block-coordinate minimization algorithm is used:

- (i) initialization $X = Y$, $D =$ overcomplete DCT dictionary;
- (ii) repeat J times:

- (1) sparse coding stage: use any pursuit algorithm to seek the optimal $\hat{\alpha}_{ij}$, by approximating the solution of

$$\forall_{ij} \min_{\alpha_{ij}} \|\alpha_{ij}\|_0 \quad \text{s.t.} \quad \|D\alpha_{ij} - R_{ij}X\|_2^2 \leq \epsilon^2; \quad (6)$$

- (2) dictionary update stage: for each column $l = 1, 2, \dots, k$ in D , and $\alpha_{ij} \neq 0$, compute the representation error

$$e_{ij}^l = R_{ij}X_{ij} - \sum_{m \neq l} d_m \alpha_{ij}(m) \quad (7)$$

and update the dictionary column \tilde{d}_l by SVD decomposition of the representation error;

(iii) Given all $\hat{\alpha}_{ij}$, update X :

$$\hat{X} = \arg \min_X \lambda \|X - Y\|_2^2 + \sum_{ij} \|D\hat{\alpha}_{ij} - R_{ij}X\|_2^2, \quad (8)$$

the simple quadratic term has a closed-form solution of the form

$$\hat{X} = \left(\lambda I + \sum_{ij} R_{ij}^T R_{ij} \right)^{-1} \left(\lambda Y + \sum_{ij} R_{ij}^T D\hat{\alpha}_{ij} \right). \quad (9)$$

2.2. Analysis Operator-B-Spline Framelet. In this subsection, we will briefly introduce the theory of frame and B-spline framelet [22]. A countable subset of $X(\Psi) = \{\psi_1, \psi_2, \dots, \psi_n\} \subset L_2(\mathbb{R})$ will be called a frame of $L_2(\mathbb{R})$, if it satisfies

$$A \|f\|^2 \leq \sum_{\psi_n \in \Psi} |\langle f, \psi_n \rangle|^2 \leq B \|f\|^2, \quad \forall f \in L_2(\mathbb{R}), \quad (10)$$

where $\langle \cdot, \cdot \rangle$ is the inner product of $L_2(\mathbb{R})$, f is a signal, and A and B are positive constants. If $A = B$, we call it tight frame. That is to say, f can be perfectly reconstructed by

$$f = \sum_{\psi_n \in \Psi} \langle f, \psi_n \rangle \psi_n, \quad \forall f \in L_2(\mathbb{R}). \quad (11)$$

Furthermore, wavelet frame is constructed by wavelet transform using time and scale parameters sampling. It is defined as

$$X(\Psi) = \left\{ \psi_{u,s} = \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right) \right\}. \quad (12)$$

If the wavelet frame satisfies the tight frame condition, functions $\psi \in \Psi$ are called framelets which belong to the wavelet tight frame $X(\Psi)$. To construct a set of framelets, one starts from a refinement mask h of a refinable function ϕ (also called a scaling function) satisfying

$$\hat{\phi}(2\cdot) = h\hat{\phi}, \quad (13)$$

where $\hat{\phi}$ is the Fourier transform of ϕ . A wavelet tight frame can be constructed by an appropriate set of framelets $\Psi = \{\psi^1, \psi^2, \dots, \psi^n\}$ that can be described in the Fourier domain as

$$\hat{\psi}^i(2\cdot) = h_i \hat{\phi}, \quad i = 1, 2, \dots, n. \quad (14)$$

B-spline tight framelet is constructed by piecewise linear B-spline function ϕ and refinement mask $h_0 = \cos^2(\omega)$. The corresponding low-pass filter is

$$\mathbf{h}_0 = \frac{1}{4} [1, 2, 1]. \quad (15)$$

Two framelets ψ_1, ψ_2 can be defined by the framelet masks $h_1 = -(\sqrt{2}i/2) \sin(\omega)$ and $h_2 = \sin^2(\omega/2)$, and the corresponding high-pass filters are

$$\mathbf{h}_1 = \frac{\sqrt{2}}{4} [1, 0, -1], \quad \mathbf{h}_2 = \frac{1}{4} [-1, 2, -1]. \quad (16)$$

Subsequently, the framelet decomposition of f can be easily obtained. The decomposition process which also can be called wavelet transform is defined as

$$Wf(u, s) = \langle f, \psi_{u,s} \rangle, \quad (17)$$

where W is the transform matrix. In our paper, W is called the analysis operator associated with 1-tight (Parseval) frame. We have $W^T W = I$, in discrete domain, and W and W^T denote the discrete wavelet tight frame decomposition and reconstruction, respectively. More details about the properties of frame and framelet can be found in [23].

3. Sparse and Redundant Representation on Dictionary Learning and Framelet

3.1. Proposed Model. Inspired by K-SVD for dictionary learning and the split Bregman method, we propose a minimization model based on the sparse representation of the image over a certain unknown dictionary and analysis prior. The proposed model can be described by

$$\begin{aligned} \min_{\alpha, u} \langle 1, Hu - g \log Hu \rangle + \lambda \|Wu\|_1 + \tau \sum_{ij} \|R_{ij}u - D\alpha_{ij}\|_2^2 \\ + \tau \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_1, \end{aligned} \quad (18)$$

where λ, τ are the positive regularization parameters. Among these parameters, τ is an important parameter. It can reduce the workload of parameter setting drastically. The effect can be seen in parameter setting analysis at the end of this section. $\|\cdot\|_1$ is the l_1 -norm that sums the absolute values of a vector. The sparse solution of (1) can be approximated by l_1 -norm regularizer. The l_1 -norm becomes popular in recent years. It is a more robust measurement which gives suitable penalty on flat region or edge. Besides, it fully embodies the sparsity. Most of all, it is easy to be computed. The notation $\|Wu\|_1$ stands for the analysis-based regularizer. The framelet transform W is generated by the piecewise linear B-splines framelet filter masks for its simplicity and effectiveness in image deblurring.

3.2. Alternating Split Bergman Algorithm. Split Bergman algorithm [9, 24] has shown good performance on splitting the seemingly unsolved variable optimization problem into number of easy subproblems. Considering that the general unconstrained minimization problems have the form of

$$\min_{u \in \mathbb{R}^N} \Phi(u) + \Psi(Cu), \quad (19)$$

where $C: \mathbb{R}^N \rightarrow \mathbb{R}^M$ is a bounded linear operator and both functions $\Phi: \mathbb{R}^N \rightarrow \mathbb{R} \cup \{+\infty\}$ and $\Psi: \mathbb{R}^M \rightarrow \mathbb{R} \cup \{+\infty\}$ are proper, convex, and lsc, it is assumed that (19) has a solution and $0 \in \text{int}(C \text{ dom}(\Phi) - \text{dom}(\Psi))$. To derive the split Bregman algorithm, problem (19) is transformed into the constrained minimization problem

$$\min_{u \in \mathbb{R}^N, d \in \mathbb{R}^M} \Phi(u) + \Psi(d) \quad \text{s.t.} \quad Cu = d. \quad (20)$$

Using augmented Lagrangian method, the split Bregman iteration for (19) is defined as

$$\begin{aligned} & (u^{(k+1)}, d^{(k+1)}) \\ &= \arg \min_{u \in \mathbb{R}^N, d \in \mathbb{R}^M} \left\{ \Phi(u) + \Psi(d) + \frac{1}{2\gamma} \|Cu - d\|_2^2 \right\}, \\ & b^{(k+1)} = b^{(k)} + \frac{1}{\gamma} (Cu^{(k+1)} - d^{(k+1)}). \end{aligned} \quad (21)$$

The alternating split Bregman algorithm is proposed to minimize the (21) alternatingly with respect to u and d :

$$\begin{aligned} u^{(k+1)} &= \arg \min_{u \in \mathbb{R}^N} \left\{ \Phi(u) + \frac{1}{2\gamma} \|b^{(k)} + Cu - d^{(k)}\|_2^2 \right\}, \\ d^{(k+1)} &= \arg \min_{d \in \mathbb{R}^M} \left\{ \Psi(d) + \frac{1}{2\gamma} \|b^{(k)} + Cu^{(k+1)} - d\|_2^2 \right\}, \\ b^{(k+1)} &= b^{(k)} + Cu^{(k+1)} - d^{(k+1)}. \end{aligned} \quad (22)$$

It is important to note that the penalty parameter $1/(2\gamma)$ should not be too large. If the parameter becomes very large, the minimization subproblems will become increasingly ill-conditioned, and it will lead to numerical problems [25].

If we use the alternating split Bregman algorithm directly, it will cause a little problem. Letting $d_1 = Hu$ and $d_2 = Wu$, we can obtain the minimization problem about variables u, α, D, d_1, d_2

$$\begin{aligned} & \min_{D, \alpha, d_1, d_2, u} \langle 1, d_1 - g \log d_1 \rangle + \lambda \|d_2\|_1 \\ & + \tau \sum_{ij} \|R_{ij}u - D\alpha_{ij}\|_2^2 + \tau \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_1 \\ & + \frac{1}{2\rho} \|b_1 + Hu - d_1\|_2^2 + \frac{1}{2\beta} \|b_2 + Wu - d_2\|_2^2. \end{aligned} \quad (23)$$

Supposing that D, α, d_1, d_2 are known, the solution with respect to u can be transformed to

$$\begin{aligned} & \min_u \tau \sum_{ij} \|R_{ij}u - D\alpha_{ij}\|_2^2 + \tau \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_1 \\ & + \frac{1}{2\rho} \|b_1 + Hu - d_1\|_2^2 + \frac{1}{2\beta} \|b_2 + Wu - d_2\|_2^2. \end{aligned} \quad (24)$$

Obviously, it is a quadratic term. We can obtain the linear equation of u by its closed form

$$\begin{aligned} \hat{u} &= \left(\frac{1}{\rho} H^T (d_1 - b_1) + \frac{1}{\beta} W^T (d_2 - b_2) + \tau \sum_{ij} R_{ij}^T D \alpha_{ij} \right) \\ & \times \left(\frac{1}{\rho} H^T H + \frac{1}{\beta} W^T W + \tau \sum_{ij} R_{ij}^T R_{ij} \right)^{-1}. \end{aligned} \quad (25)$$

The value of term about variable D is large, and other elements are so small that can be ignored. Thus, \hat{u} is about the same as the following form:

$$\hat{u} \approx \frac{\sum_{ij} R_{ij}^T D \alpha_{ij}}{\sum_{ij} R_{ij}^T R_{ij}}. \quad (26)$$

In [16], it has proved that learning dictionary using K-SVD can deal with the additive noise. However, it cannot deal with the Poisson noise directly. Here, we add the special constraint $d_3 = u$ to address this difficult problem. The special constraint comes from the idea which combines the split Bregman algorithm with an alternating minimization algorithm; refer to [26]. It is shown that this special alternating optimization algorithm simplifies the complex problem. Moreover, it solves the actual problem that K-SVD algorithm is only performed on additive noise denoising or multiplicative noise denoising, and multiplicative noise can be transformed to the additive noise in logarithm domain [17].

3.3. Proposed Algorithm. The difficulties in seeking variables are nonquadratic data term and nonseparable l_1 -norm term. The alternating optimization algorithm can be used to solve these difficulties by introducing auxiliary variables $d_1 = Hu$, $d_2 = Wu$, $d_3 = u$. The minimization problem becomes

$$\begin{aligned} & \min_{\alpha, D, d_1, d_2, d_3} \langle 1, d_1 - g \log d_1 \rangle + \lambda \|d_2\|_1 \\ & + \tau \sum_{ij} \|R_{ij}d_3 - D\alpha_{ij}\|_2^2 + \tau \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_1 \\ & \text{s.t. } d_1 = Hu, \quad d_2 = Wu, \quad d_3 = u. \end{aligned} \quad (27)$$

Here, the alternating optimization algorithm is a special split Bregman algorithm. It is to be noted that d_3 replaced u in the sparse representation of the image over a certain unknown dictionary.

The proposed model is finally defined as

$$\begin{aligned} & \min_{\alpha, D, u, d_1, d_2, d_3} \langle 1, d_1 - g \log d_1 \rangle + \lambda \|d_2\|_1 \\ & + \tau \sum_{ij} \|R_{ij}d_3 - D\alpha_{ij}\|_2^2 + \tau \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_1 \\ & + \frac{1}{2\rho} \|b_1 + Hu - d_1\|_2^2 + \frac{1}{2\beta} \|b_2 + Wu - d_2\|_2^2 \\ & + \frac{1}{2\gamma} \|b_3 + u - d_3\|_2^2, \end{aligned} \quad (28)$$

where ρ, β, γ are the positive penalty parameters. The minimization problem (28) can be decoupled into several subproblems via split Bregman iteration.

(1) Given $d_1^{(k)}, d_2^{(k)}, d_3^{(k)}, b_1^{(k)}, b_2^{(k)}, b_3^{(k)}$, the minimization of (28) with respect to u is transformed to

$$\begin{aligned} \min_u \frac{1}{2\rho} \|b_1^{(k)} + Hu - d_1^{(k)}\|_2^2 + \frac{1}{2\beta} \|b_2^{(k)} + Wu - d_2^{(k)}\|_2^2 \\ + \frac{1}{2\gamma} \|b_3^{(k)} + u - d_3^{(k)}\|_2^2. \end{aligned} \quad (29)$$

According to its closed form, we can obtain the linear equation of $u^{(k+1)}$

$$\begin{aligned} u^{(k+1)} \\ = \left(\frac{1}{\rho} H^T (d_1^{(k)} - b_1^{(k)}) + \frac{1}{\beta} W^T (d_2^{(k)} - b_2^{(k)}) \right. \\ \left. + \frac{1}{\gamma} (d_3^{(k)} - b_3^{(k)}) \right) \times \left(\frac{1}{\rho} H^T H + \frac{1}{\beta} W^T W + \frac{1}{\gamma} \right)^{-1}. \end{aligned} \quad (30)$$

The convolution of H can be easily performed in FFT domain.

(2) Given $b_1^{(k)}, u^{(k+1)}$, the subproblem with respect to d_1 becomes

$$\min_{d_1} \langle 1, d_1 - g \log d_1 \rangle + \frac{1}{2\rho} \|b_1^{(k)} + Hu^{(k+1)} - d_1\|_2^2. \quad (31)$$

Then d_1 can be obtained via solving the corresponding Euler-Lagrange equation of functional (31)

$$\begin{aligned} d_1^{(k+1)} = \frac{1}{2} \left(b_1^{(k)} + Hu^{(k+1)} - \rho \right. \\ \left. + \sqrt{(b_1^{(k)} + Hu^{(k+1)} - \rho)^2 + 4\rho g} \right). \end{aligned} \quad (32)$$

(3) Given $b_2^{(k)}, u^{(k+1)}$, the third subproblem leads to a l_1 -norm problem

$$\min_{d_2} \lambda \|d_2\|_1 + \frac{1}{2\beta} \|b_2^{(k)} + Wu^{(k+1)} - d_2\|_2^2. \quad (33)$$

The shrinkage operator is used to solve problem (31). Supposed that $S_\lambda(t) = \text{sign}(t)(|t| - \lambda)_+$, $d_2^{(k+1)} = S_{\lambda\beta}(Wu^{(k+1)} + b_2^{(k)})_+$, $d_2^{(k+1)}$ is expanded,

$$d_2^{(k+1)} = \max \left\{ \|Wu^{(k+1)} + b_2^{(k)}\|_1 - \lambda\beta, 0 \right\} \frac{Wu^{(k+1)} + b_2^{(k)}}{\|Wu^{(k+1)} + b_2^{(k)}\|_2}. \quad (34)$$

(4) Given $u^{(k+1)}, d_3^{(k)}, D$, the sparse representation coefficients $\{\alpha_{ij}\}$ can be obtained by solving

$$\min_{\alpha_{ij}} \sum_{ij} \|R_{ij} d_3^{(k)} - D\alpha_{ij}\|_2^2 + \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_1. \quad (35)$$

Comparing (35) with (6), it can be found that (35) is the sparse coding stage of K-SVD algorithm. Any pursuit

algorithm can be used to compute the representation vectors $\tilde{\alpha}_{ij}$.

(5) Given $\tilde{\alpha}_{ij}, D, u^{(k+1)}, b_3^{(k)}$, the last subproblem with respect to d_3 is

$$\min_{d_3} \tau \sum_{ij} \|R_{ij} d_3 - D\tilde{\alpha}_{ij}\|_2^2 + \frac{1}{2\gamma} \|b_3^{(k)} + u^{(k+1)} - d_3\|_2^2. \quad (36)$$

This is also a linear equation problem; similar to (33), we can obtain

$$\begin{aligned} d_3^{(k+1)} = \left(\frac{1}{\gamma\tau} I + 2 \sum_{ij} R_{ij}^T R_{ij} \right)^{-1} \\ \times \left(\frac{1}{\gamma\tau} (b_3^{(k)} + u^{(k+1)}) + 2 \sum_{ij} R_{ij}^T D\tilde{\alpha}_{ij} \right). \end{aligned} \quad (37)$$

(6) Given $b_1^{(k)}, b_2^{(k)}, b_3^{(k)}, d_1^{(k+1)}, d_2^{(k+1)}, d_3^{(k+1)}, u^{(k+1)}$, the Bregman auxiliary variables b_1, b_2, b_3 are updated as follows:

$$\begin{aligned} b_1^{(k+1)} &= b_1^{(k)} + Hu^{(k+1)} - d_1^{(k+1)}, \\ b_2^{(k+1)} &= b_2^{(k)} + Wu^{(k+1)} - d_2^{(k+1)}, \\ b_3^{(k+1)} &= b_3^{(k)} + u^{(k+1)} - d_3^{(k+1)}. \end{aligned} \quad (38)$$

Besides, the dictionary learning is also an important procedure on sparse and redundant representation. We update the dictionary D at the last step. The D is updated in an outer loop for saving computational time as suggested in [21]. Therefore, the loop times are divided into inner loop for main process and outside loop only for dictionary updating.

Taking into account the above equations, the proposed Poissonian image deconvolution algorithm is summarized as shown in Algorithm 1.

One of the stopping criterions is calculated for each iteration by the ‘‘normalized step difference energy’’ (NSDE): $\text{NSDE} = \|u^{(k)} - u^{(k-1)}\|_2 / \|u^{(k)}\|_2^2$, where $u^{(k)}$ and $u^{(k-1)}$ denote the image vector at the k th and $(k-1)$ th iteration, respectively. The other stopping criterion is $k \leq N_I$, where N_I is the maximum inner loop times. Each of the stopping criterion is satisfied; the inner loop will be terminated. In our paper, we set $\text{NSDE} \leq 0.00001$, $N_I = 40$, and $N_O = 10$. Besides, some conditions should be satisfied to obtain a sound deconvolution result. For example, the restored image u should be normalized, and it should be nonnegative in the process of iteration as we set $u = \max(u, 0)$.

The proposed method involves many penalty parameters. Parameters choosing plays an important role in efficiency and stability of the algorithm. There are some skills to determinate these parameters. (1) In many literatures, ρ, β, γ can be equal to each other. To eliminate numerical problems, the three parameters should not be too small. (2) The value of ρ is the larger the better in practice. In our paper, we set $\rho \in [400, 10000]$. We can see that $\lambda\beta$ is an important parameters group in the process of framelet regularization. The value of group $\lambda\beta$ belongs to empirical range $[3, 10]$. The last parameters group is $\tau\gamma$ which belongs to empirical range $[1, 5]$.

```

Initialize: Initialization:  $u^0 = g, d_1^0 = Hg, d_2^0 = Wg, d_3^0 = g, D = \text{DCT}$ , and
 $b_1^0 = b_2^0 = b_3^0$ . Set  $\text{iter}_O := 0, \text{iter}_I := 0$ .
for  $\text{iter}_O = 0$  to  $N_O$  do
  while stopping criterions is not satisfied do
    Step 1. Update  $u^{(k+1)}$  using (30);
    Step 2. Update  $d_1^{(k+1)}$  using (32);
    Step 3. Update  $d_2^{(k+1)}$  using (34);
    Step 4. Update  $d_3^{(k+1)}$  using (37);
    Step 5. Update sparse representation coefficients  $\{\alpha_{ij}\}$  using (6);
    Step 6. Update the Bregman auxiliary variables  $b_1^{(k+1)}, b_2^{(k+1)}, b_3^{(k+1)}$ ;
     $\text{iter}_I = \text{iter}_I + 1$ ;
  end
  Step 7. Update Dictionary  $D$ ;
end

```

ALGORITHM 1: Poissonian image deconvolution via sparse and redundant representations and framelet regularization.



FIGURE 1: Original images. (a) brain, (b) neck, (c) Lena, (d) Bara, (e) house.

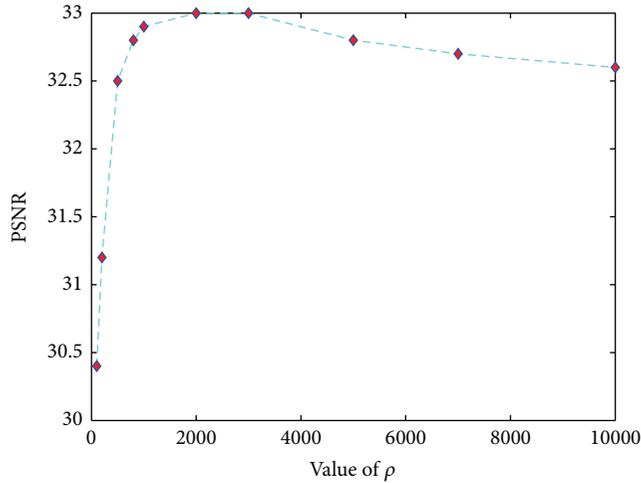


FIGURE 2: PSNR versus the value of parameter of the proposed method on Brain image contaminated by the Gaussian blur and Poisson noise with $I_{\max} = 600$.

4. Experimental Results

In this section, in order to evaluate the performance of the proposed Poissonian image deconvolution method, we test and compare it with framelet regularization based method for Poissonian image deconvolution using Bregman iteration

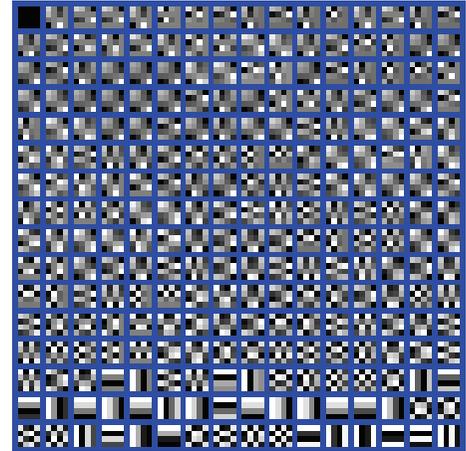


FIGURE 3: Dictionary learned by the proposed method for Example 2.

(PIDSF-FA) [13] method and dictionary learning approach and total variation regularization (TV-LD) [21] method. The ratio of peak signal and noise (PSNR) is frequently used to measure the restored image quality:

$$\text{PSNR} = 10 \log_{10} \left(\frac{L^2}{\text{NMSE}} \right), \quad (39)$$

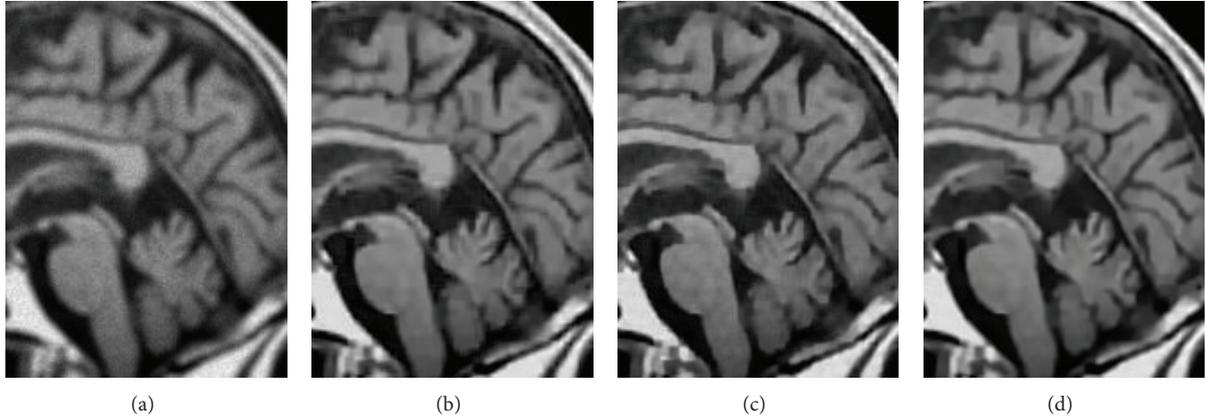


FIGURE 4: Restoration of the brain image. (a) Degraded image is blurred by uniform kernel of size 5×5 and contaminated by Poisson noise with $I_{\max} = 600$ (PSNR = 23.20 dB). (b) Restored image by the PIDSB-FA method (PSNR = 28.82 dB). (c) Restored image by the LD-TV method (PSNR = 28.08 dB). (d) Restored image by the proposed method (PSNR = 29.51 dB).



FIGURE 5: Restoration of the Bara image. (a) Degraded house image is blurred by Gaussian kernel of size 9×9 and contaminated by Poisson noise with $I_{\max} = 2000$ (PSNR = 25.14 dB). (b) Restored image by the PIDSB-FA method (PSNR = 25.68 dB). (c) Restored image by the LD-TV method (PSNR = 25.55 dB). (d) Restored image by the proposed method (PSNR = 26.29 dB).



FIGURE 6: Restoration of the Lena image. (a) Blurred by using the customized kernel and then corrupted by Poisson noise (PSNR = 26.8674 dB). (b) Restored image by the PIDSB-FA method (PSNR = 30.62 dB). (c) Restored image by the LD-TV method (PSNR = 30.66 dB). (d) Restored image by the proposed method (31.74 dB).

where L denotes the maximum value of pixel in an image, $NMSE = \frac{\|u - \hat{u}\|_2^2}{\|u\|_2^2}$, and u and \hat{u} denote the original image and restored image, respectively.

The numerical experiments are implemented with Matlab 7.14.0 on a dual core personal computer, 3.10 GHz processor and 2 G physical memory.

Example 1. To test the performance of the proposed method, numerical experiments are conducted on two kinds of test

images. The first kind of test images is medical images [21], brain image with size of 380×366 and neck image with size of 300×300 . The other kind of test images is nature images, house image with size of 300×300 , Lena image with size of 521×521 , and Bara image with size of 512×512 (Figure 1). All images are normalized in the interval $[0, 1]$.

These images are clear and original images. To simulate the Poissonian image deconvolution, these images should be

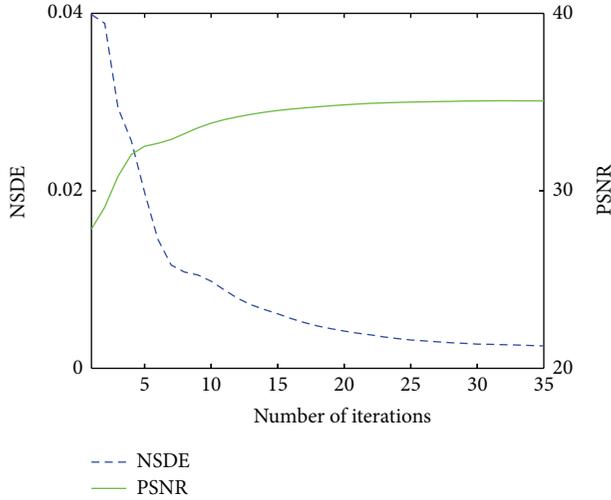


FIGURE 7: NSDE and PSNR values versus the number of iterations of the proposed method.

degraded by the blur kernel and contaminated by Poisson noise. Gaussian blur kernel and uniform blur kernel are used to test the performance of deblurring. We choose 9×9 Gaussian blur kernel with standard deviation 1 and 5×5 uniform blur kernel. Three Poisson noise levels are considered, $I_{\max} = 600, 1000, 2000$, where I_{\max} denotes the peak intensity of Poisson noise. The lower peak intensity corresponds to the stronger Poisson noise. In general, the simulated process has three steps. At first, the original image is scaled by I_{\max}/U_{\max} , where U_{\max} denotes the maximal value of the image. Subsequently, blurring process is added by convolving the normalized image with blur kernel. At last, Poisson noise is added to the blurred image.

The range of the parameters is given at the end of Section 3. In this section, we offer the detailed parameters setting. The value of parameters pair $\lambda\beta$ is 10 for all blur kernels and peak intensities of Poisson noise. We set $\tau\gamma = 3$ for Gaussian blur kernel, $\rho = 3000$, $\rho = 8000$, $\rho = 10000$ for Gaussian blur kernel, and Poisson noise with $I_{\max} = 600, 1000, 2000$, respectively. We take $\rho = 10000$, $\tau\gamma = 1$, for uniform blur kernel and all peak intensities of Poisson noise.

In all experiments, we set the parameters of K-SVD [16] as follows: iteration number is 30, the average error passing the threshold in OMP is $\epsilon = C\delta$, $C = 1.15$, $\delta = 5$, the size of dictionaries is 16×256 , and the size of image patches is 4×4 . Large patch size will produce redundant information and small patch size will lead to useful information missing. Besides, the larger image patch size is, the longer time will be taken [27]. The final dictionary learned at the last iteration of the proposed method for Example 2 is shown in (Figure 3).

To illustrate the influence of the parameter ρ on the performance of the proposed method, we note down the PSNR value of restored image when each parameter ρ is changing in (Figure 2). The plot tests and verifies the fact that parameter $1/\rho$ is not too high. Besides, the parameter receives good performance in a wide range.

The parameters setting for PIDSF-FA are offered as follows: $\gamma\lambda = 10$, $\lambda = 0.025$, and $\text{NSDE} = 10^{-4}$, and we set parameters for LD-TV as suggested in [21].

Restoration of brain is shown in Figure 4. The brain image is blurred by uniform kernel of size 5×5 and contaminated by Poisson noise with $I_{\max} = 600$. Restoration of house is shown in Figure 5. The house image is blurred by Gaussian kernel of size 9×9 and contaminated by Poisson noise with $I_{\max} = 2000$. The image restored by PIDSF-FA method contains much details, no matter useful or useless. LD-TV reduces the useless details; however, it also smooths the useful details. The proposed method gives the sound restored results. It overcomes the problems which are caused by PIDSF-FA and LD-TV methods.

The PSNR values of degraded images and restored images by the three methods are compared in Table 1. The improvement in terms of PSNR values obtained by PIDSF-FA is about the same as LD-TV. Experiment results describe that both analysis operator using piecewise linear B-splines framelet filter and synthesis operator using dictionary learning are good operators for image restoration. The proposed method combines the two operators. It can be found from Table 1 that the proposed method has achieved the highest PSNR on medical and nature images. Obviously, it remains the advantages of two operators.

Example 2. The restoration of the Lena image is blurred by customized blur ($[0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0; 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0; 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0; 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0; 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1; 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0; 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0]/22$) and then contaminated by Poisson noise with $I_{\max} = 2000$. The parameters setting for proposed method are $\rho = 10000$, $\lambda\beta = 3$, and $\tau\gamma = 1$. Restoration of the Lena image by three methods is described in Figure 6. Lena image is a widely used image for its rich repetition pattern like texture region and strong sparse representation like large flat region and smooth region. The restored image by PIDSF-FA method preserves the rich details in texture region such as decorations of the hat and also draws the artifacts into flat region such as Lena's cheek. The restored image by LD-TV is relatively more smooth in flat region; however, it also misses some fine details in texture region. Meanwhile, the proposed method performs well on highlighting the texture region and smoothing the flat region. On the other hand, the proposed method receives better PSNR values than the other two methods.

The above experiments describe that the proposed method gives better restoration of various images under different kinds of blurring kernels and Poisson noise with multilevel peak intensities. Besides, the stability is also the chief concerned matter in the proposed method. Figure 7 shows that the NSDE and PSNR value of proposed method is changing with iterations on the brain image with Gaussian blur kernel and $I_{\max} = 2000$ of Poisson noise. The plot of NSDE describes the convergence, and PSNR value of restored image increases progressively until being stable.

5. Conclusions

In this study, we proposed and implemented a Poissonian deconvolution method via sparse and redundant representations and framelet. The proposed method includes three terms: the Poisson data term, the synthesis term, and

TABLE 1: PSNR (dB) values of three methods.

Image	I_{\max}	Type	Degraded image	PIDSB-FA [13]	LD-TV [21]	Proposed method
Brain	600	Gaussian	27.06	32.19	32.73	32.80
		Uniform	23.20	28.82	28.98	29.51
	1000	Gaussian	27.48	33.33	33.67	33.94
		Uniform	23.38	29.27	29.54	29.87
	2000	Gaussian	27.84	34.33	34.35	35.07
		Uniform	23.52	29.67	29.83	31.36
Neck	600	Gaussian	26.94	30.19	30.34	30.37
		Uniform	23.62	27.21	27.35	27.50
	1000	Gaussian	27.43	33.33	33.67	33.94
		Uniform	23.82	27.43	27.48	28.12
	2000	Gaussian	27.81	31.26	31.26	32.19
		Uniform	24.00	27.57	27.66	29.19
Lena	600	Gaussian	28.84	32.89	32.81	33.45
		Uniform	27.05	31.30	31.07	32.04
	1000	Gaussian	30.02	33.76	33.63	34.08
		Uniform	27.80	31.64	31.47	31.77
	2000	Gaussian	31.18	34.34	34.30	34.88
		Uniform	28.46	31.85	31.77	32.55
Bara	600	Gaussian	24.46	25.26	25.30	25.55
		Uniform	22.65	24.50	24.41	24.54
	1000	Gaussian	24.84	25.37	25.30	25.66
		Uniform	22.89	24.56	24.49	24.57
	2000	Gaussian	25.15	25.68	25.55	26.29
		Uniform	23.09	24.59	24.57	25.04
House	600	Gaussian	27.98	32.10	32.19	32.92
		Uniform	25.88	31.08	31.13	32.04
	1000	Gaussian	29.02	32.73	33.00	33.66
		Uniform	26.49	31.35	31.63	32.35
	2000	Gaussian	30.02	33.34	33.68	34.53
		Uniform	27.03	31.53	31.99	33.09

the analysis term. Dictionary learning for synthesis model and B-spline framelet-based analysis prior are integrated perfectly. The proposed method keeps the satisfied result of dictionary learning method; meanwhile, it takes full advantage of the multiscale characteristic of framelet-based methods. The seemingly unsolved variable optimization problem is efficiently addressed by using the specific split Bregman method. Furthermore, parameters are divided into groups which simplify the parameter setting greatly. Numerical comparisons on a various of simulated images show that the proposed method outperforms the other two latest methods. However, the dictionary learning is a time-consuming process. Therefore, improving the proposed method on dictionary learning in terms of computation time is the future research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

Yu Shi and Houzhang Fang contributed equally to this work.

References

- [1] S.-M. Chao and D.-M. Tsai, "Astronomical image restoration using an improved anisotropic diffusion," *Pattern Recognition Letters*, vol. 27, no. 5, pp. 335–344, 2006.
- [2] V. Y. Panin, G. L. Zeng, and G. T. Gullberg, "Total variation regulated EM algorithm," *IEEE Transactions on Nuclear Science*, vol. 46, no. 6, pp. 2202–2210, 1999.
- [3] Y. Wang, Q. Dai, Q. Cai, P. Guo, and Z. Liu, "Blind deconvolution subject to sparse representation for fluorescence microscopy," *Optics Communications*, vol. 286, pp. 60–68, 2013.
- [4] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, 2007.
- [5] N. Dey, L. Blanc-Feraud, C. Zimmer et al., "Richardson-Lucy algorithm with total variation regularization for 3D confocal

- microscope deconvolution," *Microscopy Research and Technique*, vol. 69, no. 4, pp. 260–266, 2006.
- [6] L. Yan, H. Fang, and S. Zhong, "Blind image deconvolution with spatially adaptive total variation regularization," *Optics Letters*, vol. 37, no. 14, pp. 2778–2780, 2012.
- [7] A. Sawatzky, C. Brune, F. Wiibbeling, T. Kusters, K. Schäfers, and B. Martin, "Accurate EM-TV Algorithm in PET with Low SNR," in *Proceedings of the IEEE Nuclear Science Symposium Conference Record (NSS/MIC '08)*, pp. 5133–5137, October 2008.
- [8] S. Setzer, G. Steidl, and T. Teuber, "Deblurring Poissonian images by split Bregman techniques," *Journal of Visual Communication and Image Representation*, vol. 21, no. 3, pp. 193–199, 2010.
- [9] J.-F. Cai, S. Osher, and Z. Shen, "Split Bregman methods and frame based image restoration," *Multiscale Modeling & Simulation*, vol. 8, no. 2, pp. 337–369, 2009/10.
- [10] D.-Q. Chen and L.-Z. Cheng, "Deconvolving poissonian images by a novel hybrid variational model," *Journal of Visual Communication and Image Representation*, vol. 22, no. 7, pp. 643–652, 2011.
- [11] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3133–3145, 2010.
- [12] H. Fang, L. Yan, H. Liu, and Y. Chang, "Blind Poissonian images deconvolution with framelet regularization," *Optics Letters*, vol. 38, no. 4, pp. 389–391, 2013.
- [13] H. Fang and L. Yan, "Poissonian image deconvolution with analysis sparsity priors," *Journal of Electronic Imaging*, vol. 22, no. 2, pp. 023033-1–023033-10, 2013.
- [14] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: a dictionary-learning algorithm for the analysis sparse model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 661–677, 2013.
- [15] Y. Xiao and T. Zeng, "Poisson noise removal via learned dictionary," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 1177–1180, September 2010.
- [16] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [17] Y.-M. Huang, L. Moisan, M. K. Ng, and T. Zeng, "Multiplicative noise removal via a learned dictionary," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4534–4543, 2012.
- [18] M. Mäkitalo and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising," *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 99–109, 2011.
- [19] M. Carlván and L. Blanc-Féraud, "Sparse Poisson noisy image deblurring," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1834–1846, 2012.
- [20] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013.
- [21] L. Ma, L. Moisan, J. Yu, and T. Zeng, "A dictionary learning approach for Poisson image deblurring," *IEEE Transactions on Medical Imaging*, vol. 32, no. 7, pp. 1277–1289, 2013.
- [22] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "Framelet-based blind motion deblurring from a single image," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 562–572, 2012.
- [23] A. Ron and Z. Shen, "Affine systems in $L_2(\mathbb{R}^d)$: the analysis of the analysis operator," *Journal of Functional Analysis*, vol. 148, no. 2, pp. 408–447, 1997.
- [24] T. Goldstein and S. Osher, "The split Bregman method for L_1 -regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.
- [25] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, 2010.
- [26] Y. Huang, M. K. Ng, and Y.-W. Wen, "A fast total variation minimization method for image restoration," *Multiscale Modeling & Simulation*, vol. 7, no. 2, pp. 774–795, 2008.
- [27] J. F. Cai, H. Ji, Z. Shen, and G. B. Ye, "Data-driven tight frame construction and image denoising," *Applied and Computational Harmonic Analysis*. In Press, <ftp://ftp.math.ucla.edu/pub/camreport/caml2-40.pdf>.

Research Article

Rendering Distortion Estimation Model for 3D High Efficiency Depth Coding

Qiuwen Zhang,¹ Liang Tian,² Lixun Huang,¹ Xiaobing Wang,³ and Haodong Zhu¹

¹ College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

² College of Computer and Information Engineering, Xinxiang College, Xinxiang 453003, China

³ Technology & Information Center, CPI Henan Power Limited Company, Zhengzhou 450001, China

Correspondence should be addressed to Qiuwen Zhang; zhangqwen@126.com

Received 23 September 2013; Accepted 23 December 2013; Published 16 January 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Qiuwen Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A depth map represents three-dimensional (3D) scene geometry information and is used for depth image based rendering (DIBR) to synthesize arbitrary virtual views. Since the depth map is only used to synthesize virtual views and is not displayed directly, the depth map needs to be compressed in a certain way that can minimize distortions in the rendered views. In this paper, a modified distortion estimation model is proposed based on view rendering distortion instead of depth map distortion itself and can be applied to the high efficiency video coding (HEVC) rate distortion cost function process for rendering view quality optimization. Experimental results on various 3D video sequences show that the proposed algorithm provides about 31% BD-rate savings in comparison with HEVC simulcast and 1.3 dB BD-PSNR coding gain for the rendered view.

1. Introduction

3D video has gained increasing interest recently. It provides viewers with the illusion of 3D depth perception. The typical 3D video is represented using the multiview video plus depth (MVD) format [1, 2], in which few captured texture videos as well as associated depth maps are used. The depth maps provide per-pixel with depth corresponding to the texture video that can be used to render arbitrary virtual views by using depth image based rendering (DIBR) [3, 4]. For such depth enhanced 3D formats, high efficiency 3D video coding solutions are currently being developed in joint collaborative team on 3D video coding extension development (JCT-3V).

Since depth enhanced 3D video MVD representation causes huge amount of data to be stored or transmitted, it is essential to develop efficient 3D video coding techniques. The most straightforward approach to compress 3D video is using conventional video compression algorithms. The next generation video compression standard HEVC developed by Joint video team (JVT) is joined by both ISO/IEC motion picture experts (MPEG) and ITU-T video coding experts group (VCEG). It provides about 50% bit rate reduction as compared to H.264/AVC achieving the same subjective video

quality [5]. Therefore even simulcast HEVC compression of multiview video is more efficient than multiview video coding (MVC). For 3D video coding, a simple extension is to apply two HEVC codecs: one for all texture videos and the other for all depth maps. However, compared to conventional 2D video images, depth maps have very different characteristics. One of the major differences is that the depth maps are only used to render virtual views but not directly used to display, so depth map coding errors cause distortions in synthesized virtual views. That is to say, in conventional texture video coding, we try to improve coding performance of video data; however, in depth maps coding, we focus more on better rendering quality rather than on depth quality. Thus, use of existing HEVC codecs to compress depth maps will introduce distortions into the novel virtual views.

To solve this problem, several approaches have been proposed to enhance the depth coding performance of synthesized view quality. Kim et al. [6] analyzed the geometry error in synthesized view due to depth map lossy coding, Oh et al. [7] introduced a novel distortion function to measure block distortion for synthesized view, and De Silva and Fernando [8] optimized intramode selection for depth map coding to

minimize rendering distortions only for a single viewpoint. Although these methods have better performance than the MVC depth coding methods, they are not compatible with the HEVC standards. In order to meet the compatibility of the HEVC standard, depth maps coding algorithms have moved interest on reducing compression artifacts that may exist in depth maps which are encoded by HEVC. A new distortion metric [9] base on HEVC was used for depth map coding to replace the conventional distortion function for 3D video coding. However, the method could not completely reflect the virtual view rendering process, so that the depth coding performances are not satisfactory.

In order to optimally solve the rendering distortion model problem in MVD coding with high accuracy, a novel distortion model is proposed to precisely estimate distortion in view rendering in this paper. Compared with the previous work for view rendering distortion model, the proposed distortion model highly improves the coding efficiency with carefully considering the depth error sensitivity and occlusion handling with low complexity. The modified distortion model gives more optimal decisions for tree block rate distortion optimization with regard to rendering view quality, based on HEVC technology. Experimental results are given to demonstrate the higher performance of the proposed new rendering distortion estimation algorithm.

2. Proposed Rendering Distortion Estimation

In this section, we derive a relationship between coding errors in the depth map and geometry errors in view rendering and propose a distortion estimation model for view synthesis caused by depth map compression.

Depth image based rendering (DIBR) is the process of synthesizing virtual views of a scene from reference color images and associated per-pixel depth information:

$$\begin{aligned} & [x', y', z']^T \\ &= A_v \cdot R_v^{-1} \cdot \{R_r \cdot A_r^{-1} \cdot [x, y, 1]^T \cdot Z_r(x, y) + T_r - T_v\}, \end{aligned} \quad (1)$$

where A_r , R_r , and T_r , respectively, represent the reference camera parameters of the intrinsic matrices, rotation matrix, and translation vector, respectively. $Z_r(x, y)$ is the depth value associated with (x_r, y_r) , the subscript v refers to the virtual view, and the subscript r indicates the reference view. The corresponding pixel located in the rendered image of the virtual view is $(x_v, y_v) = (x'/z', y'/z')$. According to (1), an arbitrary virtual view can be generated, when the depth value $Z_r(x_r, y_r)$ is known for every pixel in the reference image and the camera parameters are available.

In 3D video applications, efficient compression of texture video and depth map are necessary. Due to the strict limitation of data rate in 3D video broadcast, only the lossy compression of the texture video and depth map can meet the bandwidth requirement. During HEVC coding, texture video and depth map values are subject to coding errors, such that their reconstructed values differ from the original. While texture video errors only change the interpolation value,

according to (1), per-pixel depth value determines how much the corresponding color pixel needs to be shifted when the virtual views are rendered. The depth map error will lead to a geometric error in the interpolation, which will cause view synthesis artifacts. Coding errors in depth map cause artifacts in rendering views, as explained in more detail below.

For 3D video systems with a horizontal camera arrangement, the view synthesis can be carried out using displacement of the original camera views towards the new spatial positions in the intermediate views. These shift values are derived from the depth data. The pixel value in depth map v represents depth Z at pixel location (x, y) , and we have

$$v(x, y) = 255 \cdot \frac{1/Z(x, y) - 1/Z_{\text{far}}}{1/Z_{\text{near}} - 1/Z_{\text{far}}}, \quad (2)$$

where Z_{near} and Z_{far} are the nearest and farthest depth values in the scene, which correspond to values 255 and 0 in the depth map v .

For a horizontal camera arrangement, the depth values in different camera coordinates will be approximately equal to the depth values in the world coordinate Z . Thus the view warping in (1) can be simplified as

$$\begin{aligned} [x_v, y_v, 1]^T &= \left[\frac{x'}{z'}, \frac{y'}{z'}, \frac{z'}{z'} \right]^T \\ &= A_v \cdot R_v^{-1} \cdot R_r \cdot A_r^{-1} \cdot [x_r, y_r, 1]^T \\ &\quad + \frac{1}{Z_r(x_r, y_r)} \cdot A_v \cdot R_v^{-1} \cdot \{T_p - T_{p'}\}. \end{aligned} \quad (3)$$

In depth map coding, the quantization brings the depth map distortion. To examine the influence of depth map compression on synthesis quality, we approximate the coding effect of depth map by an additive $\Delta Z_r(x_r, y_r)$, which can be represented as

$$\widehat{Z}_r(x_r, y_r) = Z_r(x_r, y_r) + \Delta Z_r(x_r, y_r), \quad (4)$$

where $\widehat{Z}_r(x_r, y_r)$ is the compressed depth map.

Through the 3D warping, the depth distortion further results in warping error in the synthesized view image. The depth error $\Delta Z_r(x_r, y_r)$ causes the projection of the pixel p moving from p' to \widehat{p}' and results in geometry distortion:

$$\begin{aligned} \widehat{p}' &= p' + \Delta p' = [x_v + \Delta x_v, y_v + \Delta y_v, 1]^T \\ &= \left[\frac{x' + \Delta x'}{z' + \Delta z'}, \frac{y' + \Delta y'}{z' + \Delta z'}, 1 \right]^T. \end{aligned} \quad (5)$$

With (4) and (5), the virtual view synthesis after depth coding can be represented as

$$\begin{aligned}
& [x_v + \Delta x_v, y_v + \Delta y_v, 1]^T \\
&= \left[\frac{x' + \Delta x'}{z' + \Delta z'}, \frac{y' + \Delta y'}{z' + \Delta z'}, 1 \right]^T \\
&= A_v \cdot R_v^{-1} \cdot R_r \cdot A_r^{-1} \cdot [x, y, 1]^T \\
&\quad + \frac{1}{Z_r(x_r, y_r) + \Delta Z_r(x_r, y_r)} \cdot A_v \cdot R_v^{-1} \cdot \{T_r - T_v\}.
\end{aligned} \tag{6}$$

The rendering position error can be calculated by subtracting (6) from (3) as

$$\begin{aligned}
& [\Delta x_v, \Delta y_v, 1]^T \\
&= \left(\frac{1}{Z_r(x_r, y_r) + \Delta Z_r(x_r, y_r)} - \frac{1}{Z_r(x_r, y_r)} \right) \\
&\quad \cdot A_v \cdot R_v^{-1} \cdot \{T_r - T_v\}.
\end{aligned} \tag{7}$$

The per-pixel depth value $Z_r(x_r, y_r)$ can be calculated from $v_r(x_r, y_r)$, using (2):

$$\frac{1}{Z_r(x_r, y_r)} = \frac{v_r(x_r, y_r)}{255} \cdot \left(\frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}} \right) + \frac{1}{Z_{\text{far}}}. \tag{8}$$

The derivative of (9) can be calculated by combining (7) and (8):

$$\begin{aligned}
& [\Delta x_v, \Delta y_v, 1]^T \\
&= \frac{\Delta v_r(x_r, y_r)}{255} \cdot \left(\frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}} \right) \cdot A_v \cdot R_v^{-1} \cdot \{T_r - T_v\}.
\end{aligned} \tag{9}$$

The linear relationship between the depth map distortion Δv and the rendering position error $\Delta p'$ in the rendered view can be represented as

$$\Delta p'(\Delta x_v, \Delta y_v) = k \cdot \Delta v_r(x_r, y_r) \tag{10}$$

$$k = \frac{1}{255} \cdot \left(\frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}} \right) \cdot A_v \cdot R_v^{-1} \cdot \{T_r - T_v\}, \tag{11}$$

where Δx_v is the horizontal error, Δy_v the vertical error, $\Delta v_r(x_r, y_r)$ is the depth map distortion at the reference camera position p , and k is the scale factor determined by the camera parameters and the depth ranges as shown in (11). We set up a horizontal camera arrangement, which can be used for a horizontal arc camera system. Since 3D video uses parallel camera setups, the view synthesis can be carried out only using horizontal displacement of the reference camera views towards the new positions in the intermediate views. The disparity or the relative shift generated by the DIBR

algorithm is only in the horizontal direction; hence the calculation in vertical direction can be omitted. Thus, (10) can be simplified when Δy_v equals 0:

$$\Delta p'(\Delta x_v) = k \cdot \Delta v_r(x_r, y_r). \tag{12}$$

For example, if the distance between cameras T_r and T_v is large or the camera captures a near object, $1/Z_{\text{near}}$ becomes large; that is, k will be large, so that the geometry error will increase. This indicates that dense camera setting and farther scene composition are more robust to depth coding distortion in the rendering process.

Since HEVC encoding algorithms operate block-based structure, the calculation of depth map distortion to the view rendering distortions must be block-based as well. Therefore, it needs to be extended to calculate the exact view rendering distortions for a region based distortion estimation, which will be studied in more detail below.

In DIBR, complex textured images without depth discontinuities and object boundaries are very sensitive regions, and images with less textures or depth discontinuities are less sensitive. Impact of geometry error caused by depth error on view synthesis distortion depends on local characteristics of the images. To more precisely estimate the view synthesis distortion caused by the depth compression error, we propose to use the reference texture image that belongs to the same viewpoint as the depth map. Geometry error will have minimal impact on region with less textures. On the other hand, in region with object boundaries and complex texture, small changes in position can lead to significant changes in view synthesis. Thus, we classify a reference video image into several regions according to the local video characteristics. To define the area of supporting for each view synthesis distortion modeling function, we employ a quadtree decomposition to divide the reference video image into blocks of variable size. In each region, the depth values of all pixels are almost the same. The geometry errors of all pixels are almost constant in one region. Therefore, each region of the video image can be approximated by one view synthesis distortion modeling function. Due to the similarity between warping error and motion vector, the spectrum distortion analysis approach proposed in [10] is adopted to calculate the geometry error induced distortion D_{R_k} of each region R_k , which is expressed as

$$D_{R_k} = \|\Delta p'\|^2 \cdot \psi_{R_k} = \|k \cdot \Delta v_r(x_r, y_r)\|^2 \cdot \psi_{R_k}, \tag{13}$$

where $\Delta p'$ is the rendering position error calculated in (10). ψ_{R_k} represents the motion sensitivity factor of the region R_k in reference video image, which is computed as

$$\psi_{R_k} = \frac{1}{2 \cdot (2\pi)^2} \cdot \iint_{(-\pi, \pi)} S_x(\omega_1, \omega_2) \cdot (\omega_1^2 + \omega_2^2) d\omega_1 d\omega_2, \tag{14}$$

where $S_x(\omega_1, \omega_2)$ denotes the energy density of the region R_k in warping reference frame and (ω_1, ω_2) are the two-dimensional frequency vectors. Since the rendering position error is linear to the depth coding error, the synthesis distortion of region R_k can be computed as (13).

Besides, geometry occlusion and disocclusion varying in warping process also cause a significant variance in view rendering distortion. In DIBR, to inpaint the occlusion regions, the disocclusion pixels located near to the occlusion pixels are used for inpainting, which also contributes to the view rendering distortion. Though the holes due to occlusion regions are not very big (when current multiview camera array is set with a very small baseline, the occlusion regions are very small and the occlusion distortion can be tiny), the view rendering distortion caused by occlusion cannot be neglected. For the view rendering distortion in (13) did not consider holes or occlusion regions in warping process, we add the occlusion handling to improve the accuracy of the view rendering distortion estimation.

If a pixel will be occluded after warping with its neighboring view image, the depth value of this pixel would not be as important as the pixels in disocclusion regions, and according to this condition we could adjust the proposed view rendering distortion estimation model. Then, (13) can be rewritten as follows:

$$D_{R_k} = \begin{cases} (1 - \alpha)^2 \cdot \|k_L \cdot \Delta v_L(x_L, y_L)\|^2 \cdot \psi_L \\ \quad + \alpha^2 \cdot \|k_R \cdot \Delta v_R(x_R, y_R)\|^2 \cdot \psi_R, \\ \quad \text{if } R_k \text{ is visible in both cameras } L \text{ and } R, \\ \|k_L \cdot \Delta v_L(x_L, y_L)\|^2 \cdot \psi_L, \\ \quad \text{if } R_k \text{ is only visible in camera } L, \\ \|k_R \cdot \Delta v_R(x_R, y_R)\|^2 \cdot \psi_R, \\ \quad \text{if } R_k \text{ is only visible in camera } R, \\ D_o, \\ \quad \text{otherwise,} \end{cases} \quad (15)$$

$$\alpha = \frac{|T_V - T_L|}{|T_V - T_L| + |T_V - T_R|}, \quad (16)$$

where D_o represents the rendering distortion induced by inpainting on the occlusion regions of the rendered virtual view image. The subscript L refers to the left camera and the subscript R indicates the right camera. α is a weighting coefficient defined in (16). T_L , T_R , and T_V are the translation vectors of the left camera, right camera, and virtual camera, respectively.

In occlusion regions, disocclusion pixels located near to the occlusion pixels are used for inpainting. Because the original obtained pixel is not available, we assume that the distribution of pixel i_o of occlusion regions is D_{i_o} . The occlusion rendering distortion D_{i_o} is estimated by the disocclusion pixels located near to i_o , and the pixel i_o will be filled with value $H(i_o)$, which is determined by the hole filling method employed. Accordingly, D_{i_o} can be calculated as follows:

$$D_{i_o} = |V(i_o) - H(i_o)|, \quad (17)$$

where $V(i_o)$ represents the luma value at pixel i_o in virtual texture image. Consequently, the distortion D_o caused by

inpainting on the occlusion regions can be calculated as follows:

$$D_o = \sum_{i_o} |V(i_o) - H(i_o)|. \quad (18)$$

Finally, the overall view rendering distortion D_{R_k} can be estimated by (15) and (18). This model provides a pixel wise approximation to rendering errors, and these errors need to be minimized in HEVC depth coding.

3. Rate Distortion Optimization for HEVC Encoder

To enable rate distortion (RD) optimization using the proposed rendering distortion estimation, the described rendering distortion model is integrated in HEVC depth maps coding. As a 2D representation of the 3D scene surface, depth maps are utilized for rendering virtual views, and they will not be directly displayed. The quality of decoded depth map has only limited practical meaning. Therefore, the impact of HEVC depth coding artifacts needs to be evaluated further with respect to the rendering quality of virtual views. For this, the HEVC distortion computation carried out is replaced with the proposed rendering distortion estimation in all distortion computation steps. HEVC encoding mode decision is taken in a way that minimizes the errors in the image rendered by the depth map; in contrast to minimizing errors in the depth map itself, the computation of RD cost J has been modified to

$$J = D_v + \varphi \cdot \lambda \cdot R_d, \quad (19)$$

where D_v denotes the virtual view synthesis distortion caused by the HEVC depth compression, as provided by the proposed rendering distortion estimation in Section 2, φ is the constant scaling factor, R_d is the rate of the encoding depth map, and λ is the Lagrange multiplier in the HEVC encoder. We apply the new distortion D_v to the RD optimized mode selection process to decide whether the proposed prediction mode is used for current tree block. That is, when the Lagrange cost is calculated in the HEVC encoder, the estimated distortion in the rendered view is used in depth map coding.

4. Experimental Results

In order to verify the effectiveness of the proposed rendering distortion estimation algorithm, we implemented it into 3DV HEVC test model version 3.0. For the experiments, 8 MVD test sequences of various resolutions with different signal characteristics are used. Four of them were in the 1024 × 768 resolution (Kendo [11], Balloons [11], Lovebird1 [12], and Newspaper [13]) with 30 fps. Other 4 test sequences were in HD resolution of 1920 × 1088 (Undo-Dancer [14], GT-Fly [15], Poznan-Street [16], and Poznan-Hall2 [16]) with 25 fps. The detailed information of the test sequences is provided in Table 1. All 8 test sequences were evaluated in the two view cases. Depth maps were encoded using context-based

TABLE 1: Test sequence information.

Sequence	Resolution	Frame rate	GOP	Views
Kendo	1024 × 768	30	15	1-3-5
Balloons	1024 × 768	30	15	1-3-5
Lovebird1	1024 × 768	30	15	4-6-8
Newspaper	1024 × 768	30	15	2-4-6
Undo_Dancer	1920 × 1088	25	12	1-5-9
GT_Fly	1920 × 1088	25	12	9-5-1
Poznan_Street	1920 × 1088	25	12	5-4-3
Poznan_Hall2	1920 × 1088	25	12	7-6-5

TABLE 2: Encoder settings for test.

Codec	3D-HTM ver. 3.0
Number of frames	Full length
Interview prediction	P-I-P
Motion search range	64
MaxCU size	64 × 64
Depth QP	25, 30, 35, and 40
Configuration	Random access
View synthesis	VSRS 3.5

adaptive binary arithmetic entropy coding (CABAC) entropy coding and temporal prediction structures with hierarchical B-frames with GOP of 12 for 1024 × 768 resolution test sequences and 15 for 1024 × 768 resolution test sequences. The experimental results have been conducted using the test conditions of the MPEG 3DV standardization [17]. The more detailed encoder setting is provided in Table 2.

The proposed rendering distortion estimation algorithm is compared with the newly adopted 3D HEVC test model (3D-HTM ver.3.0) [18], the HEVC extension to multiple views (MV-HEVC ver.3.0) [18], and the HEVC simulcast (HM ver.6.0) [19] in terms of average PSNR and bit rate savings. The PSNR is calculated for the virtual views between the decoded synthesized views and the synthesized virtual views using uncompressed texture video and depth map. The reconstructed depth map and texture video (texture videos were not encoded) were used as inputs of view synthesis performed by using MPEG view synthesis reference software (VSRS) [20].

The rate distortion performance comparison of the proposed algorithm compared with 3D-HTM, MV-HEVC, and HEVC simulcast algorithm is shown in Figure 1. The horizontal and vertical axes represent the depth map bit rate and the quality of the rendering virtual view, respectively. From Figure 1, we can see that the proposed rendering distortion estimation algorithm is more effective than the distortion estimation algorithm in 3D-HTM, MV-HEVC, and HEVC simulcast.

Table 3 gives the coding performance of the proposed algorithm compared with 3D-HTM, MV-HEVC, and HEVC

simulcast algorithm. Bitrate (BDBR) [21] represents the improvement of total bitrates for depth map coding; Bjontegaard Delta PSNR (BD-PSNR) represents the average PSNR gain over all coded virtual views rendering. From Table 3, we can observe that, compared with HEVC simulcast, a maximum BD-rate of 48.71% can be achieved for the depth map of “GT_Fly” and the average BD-rate brought by the proposed method is 31.20%, while the average BD-PSNR increase is 1.337 dB. Compared with MV-HEVC, the proposed algorithm performs better on all the sequences and achieves more than 20.66% coding bitrate saving, with a maximum of 24.5% in “GT_Fly” and a minimum of 14.6% in “Undo_Dancer.” Meanwhile, the average PSNR increase for all the test sequences is 0.846 dB. Moreover, the proposed algorithm shows better performance with about 12.90% BD-rate gain and 0.517 dB BD-PSNR increasing than those achieved by 3D-HTM algorithm. Comparing the previous algorithms, the proposed distortion model algorithm fully mimics the warping view rendering process by even considering the depth sensitivity and occlusion process. Furthermore, the proposed RD cost function gives more optimal decision for tree block mode selection with regard to rendering view quality. Experimental results are given to demonstrate the significantly superior performance of the proposed new distortion model algorithm.

5. Conclusion

This paper presented a new distortion estimation model for 3D depth map compression based on the HEVC. The new distortion estimation model provides an exact measure that can be used in tree block based HEVC code. Compared with the previous work for view rendering distortion estimation, the proposed model significantly improves the coding efficiency with carefully considering the depth sensitivity and low complexity occlusion handling. Experimental results demonstrate that the derived view rendering distortion estimation is accurate and the proposed algorithm provides about 31% BD-rate savings in comparison with HEVC simulcast and 1.3 dB BD-PSNR coding gain for the rendered view.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the editors and anonymous reviewers for their valuable comments. The authors would also like to thank Microsoft Research, Nagoya University, Fraunhofer HHI, GIST, and ETRI, Poznan University, and Nokia, for providing their 3D video sequences and their valuable work on 3D video coding. This work was supported in part by the National Natural Science Foundation of China, under Grants nos. 61302118, 61374014, and 61201447 and

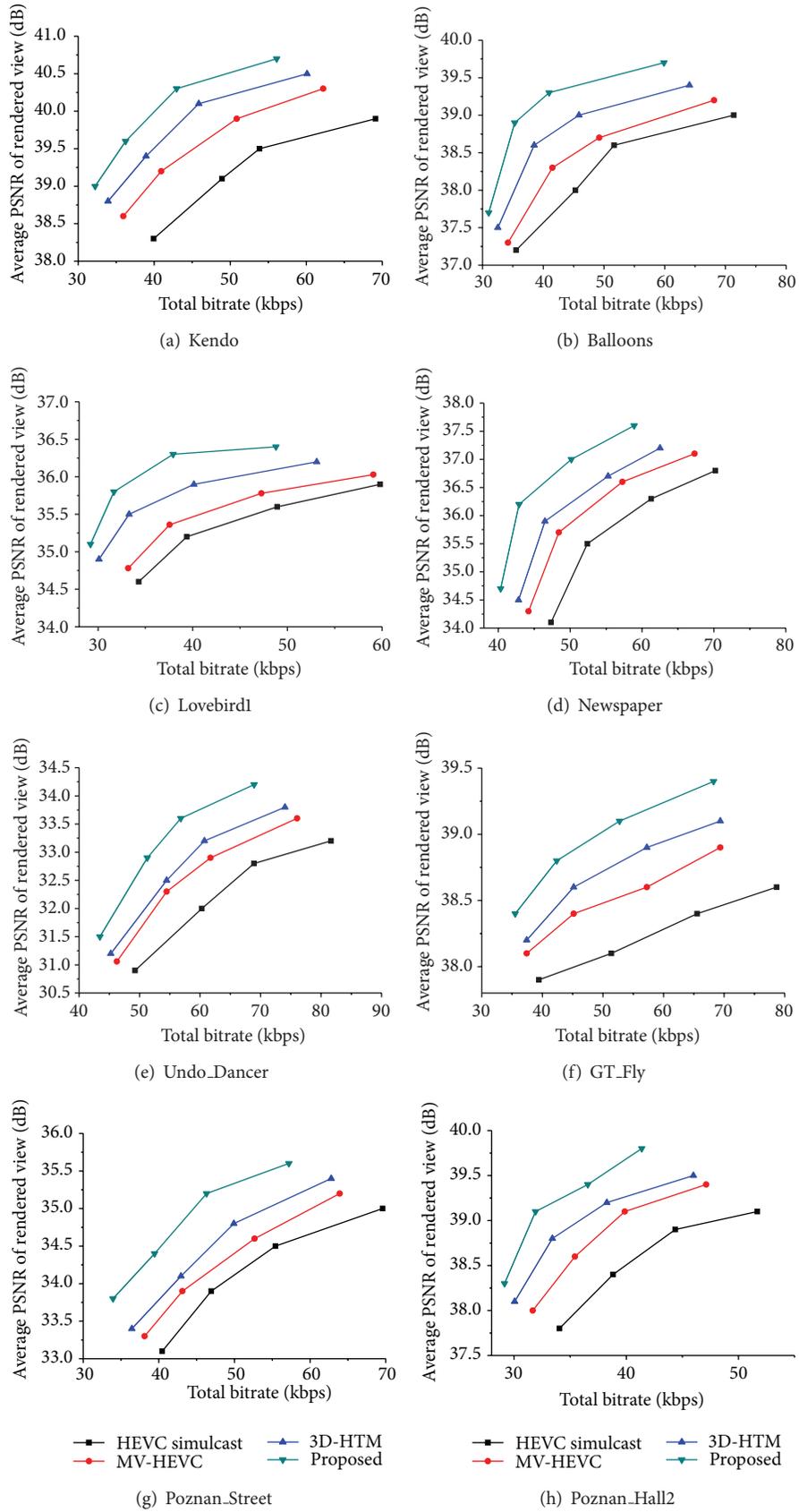


FIGURE 1: Rate distortion curves comparison.

TABLE 3: Bjontegaard delta results for proposed algorithm relative to 3D-HTM, MV-HEVC, and HEVC simulcast, comparing the decoded rendered virtual views from original uncompressed texture video and depth map.

Sequence	BD-rate (%)			BD-PSNR (dB)		
	3D-HEVC	MV-HEVC	HEVC simulcast	3D-HEVC	MV-HEVC	HEVC simulcast
Kendo	-11.955	-22.423	-35.135	0.402	0.853	1.537
Balloons	-13.231	-24.406	-31.894	0.428	0.844	1.176
Lovebird1	-8.468	-19.346	-24.397	0.444	0.806	1.052
Newspaper	-11.122	-17.092	-25.150	0.754	1.097	1.646
Undo_Dancer	-11.236	-14.624	-24.499	0.738	1.028	1.818
GT_Fly	-18.345	-24.548	-48.707	0.303	0.447	0.932
Poznan_Street	-15.651	-22.292	-28.287	0.656	0.966	1.306
Poznan_Hall2	-13.222	-20.565	-31.562	0.409	0.730	1.231
Average	-12.904	-20.662	-31.204	0.517	0.846	1.337

in part by the Doctorate Research Funding of Zhengzhou University of Light Industry, under Grant no. 2013BSJJ047.

References

- [1] K. Müller, H. Schwarz, D. Marpe, and et al., “3D high-efficiency video coding for multi-view video and depth data,” *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3366–3378, 2013.
- [2] K. Muller, P. Merkle, and T. Wiegand, “3-D video representation using depth maps,” *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, 2011.
- [3] P. Kauff, N. Atzpadin, C. Fehn et al., “Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability,” *Signal Processing: Image Communication*, vol. 22, no. 2, pp. 217–234, 2007.
- [4] C. Fehn, “Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV,” in *11th Stereoscopic Displays and Virtual Reality Systems*, vol. 5291 of *Proceedings of the SPIE*, pp. 93–104, May 2004.
- [5] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, “Comparison of the coding efficiency of video coding standards— Including high efficiency video coding (HEVC),” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, 2012.
- [6] W. S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, “Depth map coding with distortion estimation of rendered view,” in *Visual Information Processing and Communication*, vol. 7543 of *Proceeding of the SPIE*, pp. 75430B-1–75430B-10, San Jose, Calif, USA, January 2010.
- [7] B. T. Oh, J. Lee, and D.-S. Park, “Depth map coding based on synthesized view distortion function,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1344–1352, 2011.
- [8] D. V. S. X. De Silva and W. A. C. Fernando, “Intra mode selection for depth map coding to minimize rendering distortions in 3D video,” *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2385–2393, 2009.
- [9] G. Tech, H. Schwarz, K. M. Müller, and T. Wiegand, “3D video coding using the synthesized view distortion change,” in *Proceedings of the Picture Coding Symposium (PCS '12)*, pp. 25–28, Krakow, Poland, May 2012.
- [10] A. Secker and D. Taubman, “Highly scalable video compression with scalable motion coding,” *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1029–1041, 2004.
- [11] ISO/IEC JTC1/SC29/WG11, “Moving multiview camera test sequences for MPEG-FTV,” Document M16922, ISO, Xian, China, 2009.
- [12] ISO/IEC JTC1/SC29/WG11, “3D video test material of outdoor scene,” Document M15371, ISO, Archamps, France, 2008.
- [13] ISO/IEC JTC1/SC29/WG11, “Multiview video test sequence and camera parameters,” Document M15419, ISO, Archamps, France, 2008.
- [14] ISO/IEC JTC1/SC29/WG11, “Undo Dancer 3DV sequence for purposes of 3DV standardization,” Document M20028, ISO, Geneva, Switzerland, 2011.
- [15] ISO/IEC JTC1/SC29/WG11, “Ghost Town Fly 3DV sequence for purposes of 3DV standardization,” Document M20027, ISO, Geneva, Switzerland, 2011.
- [16] ISO/IEC JTC1/SC29/WG11, “Poznań multiview video test sequences and camera parameters,” Document M17050, ISO, Xian, China, 2009.
- [17] ISO/IEC JTC1/SC29/WG11, “Common test conditions for 3DV experimentation,” Document N12745, ISO, Geneva, Switzerland, 2012.
- [18] G. Tech, K. Wegner, Y. Chen, and S. Yea, *3D-HEVC Test Model*, Document JCT3V-C1005, Geneva, Switzerland, 2013.
- [19] HM6: High efficiency video coding HEVC reference software version, <https://hevc.hhi.fraunhofer.de/svn/svnHEVCSoftware/tags/HM-6.0/>.
- [20] ISO/IEC JTC1/SC29/WG11, *View Synthesis Algorithm in View Synthesis Reference Software 3.0 (VRS3.0)*, Document M16090, 2009.
- [21] G. Bjontegaard, *Calculation of Average PSNR Differences between RD-Curves*, Document VCEG-M33, 2001.

Research Article

Fast Threshold Selection Algorithm of Infrared Human Images Based on Two-Dimensional Fuzzy Tsallis Entropy

Dong-xue Xia, Chun-gui Li, and Shu-hong Yang

College of Computer Science, Guangxi University of Science and Technology, Liuzhou 545006, China

Correspondence should be addressed to Chun-gui Li; netysh2007@qq.com

Received 4 October 2013; Accepted 11 December 2013; Published 16 January 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Dong-xue Xia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Infrared images are fuzzy and noisy by nature; thus the segmentation of human targets in infrared images is a challenging task. In this paper, a fast thresholding method of infrared human images based on two-dimensional fuzzy Tsallis entropy is introduced. First, to address the fuzziness of infrared image, the fuzzy Tsallis entropy of objects and that of background are defined, respectively, according to probability partition principle. Next, this newly defined entropy is extended to two dimensions to make good use of spatial information to deal with the noise in infrared images, and correspondingly a fast computation method of two-dimensional fuzzy Tsallis entropy is put forward to reduce its computation complexity from $O(L^2)$ to $O(L)$. Finally, the optimal parameters of fuzzy membership function are searched by shuffled frog-leaping algorithm following maximum entropy principle, and then the best threshold of an infrared human image is computed from the optimal parameters. Compared with typical entropy-based thresholding methods by experiments, the method presented in this paper is proved to be more efficient and robust.

1. Introduction

Image segmentation is an important topic in the field of digital image process. It intends to extract objects from background based on some pertinent characteristics in an image such as gray level, color, texture, and location [1]. Thresholding is one of the most popular segmentation approaches because of its simplicity [2, 3]. It serves a variety of applications such as biomedical image analysis, character identification, and change detection [3]. Compared with visible images, the intensity of human targets in infrared images is obviously different from that of background. Therefore, segmenting infrared human images by threshold selection is feasible [4]. Moreover, the intensity of human targets in infrared image is mainly determined by its temperature and radiated heat and is independent of the current light conditions, so the detection system can be applied indiscriminately in both day and night [4]. However, infrared images are not perfect either. Due to the limitations in camera technology, most infrared images have lower spatial resolution and less sensitivity than visible images, which often leads to poor image quality, such as blurring, low target-to-background

contrast, and great noise. Therefore, it is a complex challenge to make precise segmentation of human targets in infrared images by threshold selection [5].

2. Related Work

Excellent reviews on early thresholding methods can be found in [6]. Among all the thresholding methods, entropy-based method is widely studied and is considered effectively [3]. Recently, Tsallis entropy was applied to the field of image thresholding for its excellent performance in the description of nonadditive information existing in images [7]. Later, this work in one-dimensional histogram was extended to two-dimensional one with much better effects [8]. Fuzzy sets play a significant role in many systems because of their capability to model nonstatistical imprecision. The notion of entropy, in the theory of fuzzy sets, was first introduced by De Luca and Termini [9]. There are numerous applications of fuzzy entropies in image segmentation. Cheng et al. proposed fuzzy homogeneity vectors to perform multilevel thresholding [10]. Tao et al. presented an approach to segmenting objects in

infrared images based on ant colony optimization algorithm and fuzzy entropy [11]. Cheng et al. presented a thresholding approach by performing fuzzy partition of a two-dimensional histogram based on fuzzy relation [12]. Zhao et al. presented an entropy function based on fuzzy c-partition (FP) and probability partition (PP) [13] to perform image thresholding.

Considering the fuzzy nature of infrared image and the excellent performance of Tsallis entropy in the description of nonadditive information, in this paper, we present a fast infrared human image thresholding method based on two-dimensional fuzzy Tsallis entropy. We first transform Tsallis entropy into fuzzy domain and then extend it to two-dimensional histogram, and thus 2-D FTE (two-dimensional fuzzy Tsallis entropy) is defined and applied to the field of threshold selection. To handle the exponential increase in computation time brought about by extending one-dimensional method to two-dimensional one, we provide a fast algorithm to reduce the computation complexity of 2-D FTE from $O(L^2)$ to $O(L)$ for an image with L gray levels. And then SHLA (shuffled frog-leaping algorithm) [14] is employed to speed up the search of optimal threshold. The comparison between the proposed method and other typical entropy-based methods demonstrates that the former is more effective and robust.

3. The Fuzzy Entropy of Images

3.1. The Fuzzy Set of Images. Let $R = \{1, 2, \dots, M\}$, $C = \{1, 2, \dots, N\}$, and $G = \{1, 2, \dots, L - 1\}$; then for an image $I = f(r, c)$ of size $M \times N$ with L gray levels, its domain can be expressed as $D = \{(r, c) \mid r \in R, c \in C\}$. Let $D_k = \{(r, c) \mid f(r, c) = k, (r, c) \in D, k \in G\}$ and denote T as a threshold by which an image is segmented into object (O) and background (B). Then $\Pi_2 = \{O, B\}$ is an unknown probability partition of D , and its probability distribution is described as $p_O = P(O)$, $p_B = P(B)$, respectively. For each $k \in G$, let

$$\begin{aligned} D_{kO} &= \{(r, c) : f(r, c) > T, (r, c) \in D_k\}, \\ D_{kB} &= \{(r, c) : f(r, c) \leq T, (r, c) \in D_k\}, \end{aligned} \quad (1)$$

then the following equations hold [11]:

$$\begin{aligned} p_{kO} &= P(D_{kO}) = p_k * p_{O|k} \\ p_{kB} &= P(D_{kB}) = p_k * p_{B|k}, \end{aligned} \quad (2)$$

where $p_{O|k}$ and $p_{B|k}$ denote the conditional probability of a pixel that is classified into the classes O and B , respectively, obviously $p_{O|k} + p_{B|k} = 1$.

To threshold an image in fuzzy domain, we first transform the image into fuzzy domain by fuzzy membership function. Suppose the fuzzy memberships of a pixel with grey level k belonging to classes O and B are $\mu_O(k)$ and $\mu_B(k)$, respectively; then $\mu_O(k) + \mu_B(k) = 1$. We can decide the class of the pixel by the value of $\mu_O(k)$ and $\mu_B(k)$. According to [11], let

$$\begin{aligned} \mu_O(k) &= p_{O|k}, \\ \mu_B(k) &= p_{B|k}, \end{aligned} \quad (3)$$

and by substituting (3) into (2), we get

$$\begin{aligned} p_O &= P(O) = \sum_{k=0}^{L-1} p_{kO} = \sum_{k=0}^{L-1} p_k * \mu_O(k), \\ p_B &= P(B) = \sum_{k=0}^{L-1} p_{kB} = \sum_{k=0}^{L-1} p_k * \mu_B(k). \end{aligned} \quad (4)$$

We select S-function and Z-function as the fuzzy membership functions because they are smoother than other fuzzy membership functions, such as Triangle function and Π -function. The standard S-function and Z-function used in this paper are shown in the following:

$$\begin{aligned} \mu_O(k) &= S(k, a, c) \\ &= \begin{cases} 0 & k \leq a \\ \frac{(k-a)^2}{(c-a)^2/2}, & a \leq k \leq \frac{a+c}{2} \\ 1 - \frac{(c-k)^2}{(c-a)^2/2}, & \frac{a+c}{2} \leq k \leq c \\ 1 & c \leq k, \end{cases} \end{aligned} \quad (5)$$

$$\mu_B(k) = Z(k, a, c) = 1 - S(k, a, c).$$

3.2. The Fuzzy Tsallis Entropy of Images. In the domain D of an image, let $h(k) = |D_k|/(N * M)$, in which $|\cdot|$ denotes the cardinality of a set. Then the histogram of an image can be expressed as $H = \{h(k) \mid k \in G\}$, and H is transformed into fuzzy domain; that is, $F = \{\mu(k)h(k) \mid k \in G\}$. Consequently, the fuzzy distribution of object O and background B can be obtained as

$$\begin{aligned} F_O &: \frac{\mu_O(0)h(0)}{p_O}, \dots, \frac{\mu_O(L-1)h(L-1)}{p_O} \\ F_B &: \frac{\mu_B(0)h(0)}{p_B}, \dots, \frac{\mu_B(L-1)h(L-1)}{p_B}, \end{aligned} \quad (6)$$

where p_O , and p_B are defined in (4).

According to the definition of Tsallis entropy, we define the fuzzy Tsallis entropy of object O and background O , respectively, as follows:

$$\begin{aligned} FT_O^q(t) &= \frac{1}{q-1} \left(1 - \sum_{l=0}^{L-1} \left(\frac{\mu_O(l)h(l)}{p_O} \right)^q \right) \\ FT_B^q(t) &= \frac{1}{q-1} \left(1 - \sum_{l=0}^{L-1} \left(\frac{\mu_B(l)h(l)}{p_B} \right)^q \right), \end{aligned} \quad (7)$$

where $q \in R^+$ and $q \neq 1$.

It can be proved that the limiting case of the fuzzy Tsallis entropy defined in this paper is equal to the fuzzy Shannon entropy defined by Tao et al. [11]. Take $\lim_{q \rightarrow 1} \text{FT}_O^q(t)$, for example, when $q \rightarrow 1$; then

$$\begin{aligned} \lim_{q \rightarrow 1} \text{FT}_O^q &= \lim_{q \rightarrow 1} \frac{1}{q-1} \left(1 - \sum_{l=0}^{L-1} \left(\frac{\mu_O(l) h(l)}{p_O} \right)^q \right) \\ &= \lim_{q \rightarrow 1} \frac{1}{(q-1)'} \left\{ 1 - \sum_{l=0}^{L-1} \left(\frac{\mu_O(l) h(l)}{p_O} \right)^q \right\}' \quad (8) \\ &= - \sum_{l=0}^{L-1} \frac{\mu_O(l) h(l)}{p_O} * \ln \left\{ \frac{\mu_O(l) h(l)}{p_O} \right\} = H_d. \end{aligned}$$

Similarly, we can prove that $\lim_{q \rightarrow 1} \text{FT}_B^q = H_b$, where H_d and H_b are the fuzzy Shannon entropy of object and background defined by Tao. Therefore, we can say that the fuzzy entropy defined by Tao is a special case of the fuzzy Tsallis entropy defined in this paper.

4. Fast Threshold Selection Based on 2D FTE

4.1. The Definition of 2-D FTE (Two-Dimensional Fuzzy Tsallis Entropy). To define the 2D FTE (fuzzy Tsallis entropy) for an image of size $M \times N$ with L gray levels, we first calculate the 2-D histogram of the image by

$$p_{ij} = \frac{n_{ij}}{(M \times N)}, \quad (9)$$

where n_{ij} denotes the number of pixels whose gray level equals i and local average gray level equals j [8]. The 2-D histogram can be viewed as a full Cartesian product of two sets X and Y , where X represents the gray levels and Y represents the local average gray levels; that is, $X = Y = \{0, 1, 2, \dots, L-1\}$. Each axis of the 2-D histogram can be transformed into fuzzy domain using S-function and Z-function, and then four fuzzy sets are defined as follows [12]:

$$\begin{aligned} OX &= \sum_{x \in X} \frac{u_{OX}}{x} = \sum_{x \in X} \frac{S(x, a, c)}{x}, \\ OY &= \sum_{y \in Y} \frac{u_{OY}}{y} = \sum_{y \in Y} \frac{S(y, a, c)}{y}, \\ BX &= \sum_{x \in X} \frac{u_{BX}(x)}{x} = \sum_{x \in X} \frac{Z(x, a, c)}{x}, \\ BY &= \sum_{y \in Y} \frac{u_{BY}(y)}{y} = \sum_{y \in Y} \frac{Z(y, a, c)}{y}. \end{aligned} \quad (10)$$

The fuzzy relation O (object) and B (background) are subsets of the full Cartesian product space $X \times Y$, respectively; that is, $O = OX \times OY \subset X \times Y$, $B = BX \times BY \subset X \times Y$. The 2-D fuzzy membership functions $\mu_O(x, y)$ and $\mu_B(x, y)$ can be obtained as follows [12]:

$$\begin{aligned} \mu_O(x, y) &= \mu_{OX \times OY}(x, y) = \min(\mu_{OX}, \mu_{OY}), \\ \mu_B(x, y) &= \mu_{BX \times BY}(x, y) = \min(\mu_{BX}, \mu_{BY}). \end{aligned} \quad (11)$$

Then mapping the 2-D histogram into fuzzy domain employing $\mu_O(x, y)$, $\mu_B(x, y)$ obtained above, we get two fuzzy distributions of classes O and B as follows:

$$\begin{aligned} \text{FP}_O &: \frac{\mu_O(0,0) h(0,0)}{p_O}, \dots, \frac{\mu_O(L-1, L-1) h(L-1, L-1)}{p_O}, \\ \text{FP}_B &: \frac{\mu_B(0,0) h(0,0)}{p_B}, \dots, \frac{\mu_B(L-1, L-1) h(L-1, L-1)}{p_B}. \end{aligned} \quad (12)$$

By the similar way used in the deducing of (4), p_O and p_B can be computed by

$$\begin{aligned} p_O &= \sum_{x=0}^{L-1} \sum_{y=0}^{L-1} \mu_O(x, y) h(x, y), \\ p_B &= \sum_{x=0}^{L-1} \sum_{y=0}^{L-1} \mu_B(x, y) h(x, y). \end{aligned} \quad (13)$$

According to the definition of two-dimensional Tsallis entropy [8], we extend the fuzzy Tsallis entropy defined in (7) to two-dimensional ones and define the 2-D FTE of classes O and B as follows:

$$\begin{aligned} \text{FH}_O^q &= \frac{1 - \sum_{x=0}^{L-1} \sum_{y=0}^{L-1} (\mu_O(x, y) h(x, y) / p_O)^q}{q-1}, \\ \text{FH}_B^q &= \frac{1 - \sum_{x=0}^{L-1} \sum_{y=0}^{L-1} (\mu_B(x, y) h(x, y) / p_B)^q}{q-1}. \end{aligned} \quad (14)$$

According to the nonadditivity rule of Tsallis entropy, the total two-dimensional fuzzy Tsallis entropy of an image can be obtained by

$$\text{FH}^q = \text{FH}_O^q + \text{FH}_B^q + (1-q) \cdot \text{FH}_O^q \cdot \text{FH}_B^q. \quad (15)$$

We follow maximum entropy principle [8] to search for the optimal threshold by

$$t_{\text{opt}} = \arg \max_{(a,c)} [\text{FH}^q]. \quad (16)$$

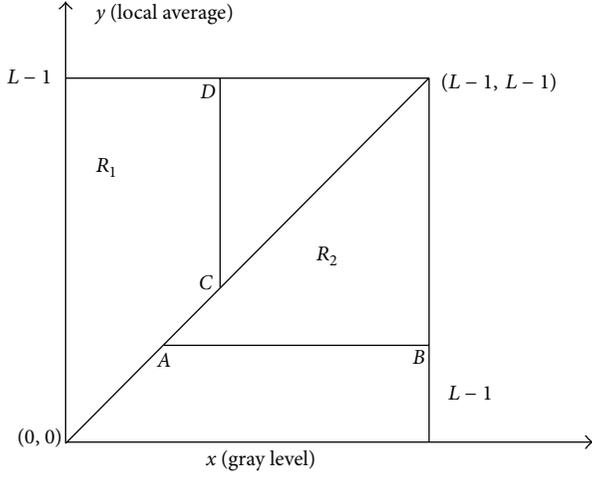


FIGURE 1: The division and accumulation of 2D histogram.

4.2. Fast Computation of Two-Dimensional Fuzzy Tsallis Entropy. The proposed segmentation method involves the optimization of (16), and thus the 2-D FTE needs to be computed once for each pair of parameters (a, c) according to (14)~(15). For an image with L gray levels, the time complexity of this computation is $O(L^2)$, which is too much even for a heuristic search algorithm like SHLA. Therefore, we present a fast algorithm to speed up the calculation. We take the derivation of the fast computation of FH_O^q as an example here; the derivation of FH_B^q is similar. We divide the 2-D histogram of image into two parts by the diagonal line from $(0, 0)$ to $(L - 1, L - 1)$ as illustrated in Figure 1.

For convenience, $S(x, a, c)$ is denoted as $S(x)$ hereafter. As S -function is a monotonically nondecreasing function, from (11), we get

$$\begin{aligned} \mu_O(x, y) &= \mu_{O_{X \times O_Y}}(x, y) = \min(\mu_{O_X}, \mu_{O_Y}) \\ &= \min(S(x), S(y)) = S(\min(x, y)). \end{aligned} \quad (17)$$

In Figure 1, $R_1 = \{(x, y), x \leq y\}$ and $R_2 = \{(x, y), x > y\}$; then, we get

$$\begin{aligned} \mu_O(x, y) &= S(\min(x, y)) = S(x), \quad (x, y) \in R_1, \\ \mu_O(x, y) &= S(\min(x, y)) = S(y), \quad (x, y) \in R_2. \end{aligned} \quad (18)$$

When the two-dimensional histogram is obtained, we first compute the four accumulation arrays, which just need to be done once for each input image:

$$\begin{aligned} U[x] &= \sum_{y=x}^{L-1} h(x, y), \quad x \in \{0, 1, \dots, L-1\}, \\ R[y] &= \sum_{x=y+1}^{L-1} h(x, y), \quad y \in \{0, 1, \dots, L-1\}, \end{aligned} \quad (19)$$

$$Up[x] = \sum_{y=x}^{L-1} [h(x, y)]^q, \quad x \in \{0, 1, \dots, L-1\}, \quad (20)$$

$$Rp[y] = \sum_{x=y+1}^{L-1} [h(x, y)]^q, \quad y \in \{0, 1, \dots, L-1\},$$

where $U[x]$ means accumulating all the elements in R_1 to the upper side of the 2-D histogram along vertical lines like CD in Figure 1. In the similar manner, $R[y]$ means accumulating all the elements in R_2 to the right side along horizontal lines like AB .

For $Up[x]$ and $Rp[y]$, we first compute $[h(x, y)]^q$ for each element of the 2-D histogram and then accumulate them to the upper and right side in the same way as $U[x]$ and $R[y]$.

For each pair of fuzzy parameters (a, c) , we first calculate $S(l)$ and $[S(l)]^q$ for each $l \in \{0, 1, \dots, L-1\}$. Then from (14) and (18), we get

$$\begin{aligned} p_O &= \sum_{(x,y) \in R_1} \mu_O(x, y) h(x, y) \\ &+ \sum_{(x,y) \in R_2} \mu_O(x, y) h(x, y) \\ &= \sum_{x=0}^{L-1} \left\{ S(x) \sum_{y=x}^{L-1} h(x, y) \right\} \\ &+ \sum_{y=0}^{L-1} \left\{ S(y) \sum_{x=y+1}^{L-1} h(x, y) \right\}. \end{aligned} \quad (21)$$

Substituting (19) into (21), we get

$$\begin{aligned} p_O &= \sum_{x=0}^{L-1} S(x) U(x) + \sum_{y=0}^{L-1} \{S(y) R(y)\} \\ &= \sum_{l=0}^{L-1} S(l) (U[l] + R[l]). \end{aligned} \quad (22)$$

In a similar manner, we get

$$\begin{aligned} &\sum_{x=0}^{L-1} \sum_{y=0}^{L-1} (\mu_O(x, y) h(x, y))^q \\ &= \sum_{l=0}^{L-1} [S(l)]^q (Up[l] + Rp[l]). \end{aligned} \quad (23)$$

Substituting (22) and (23) into (7), we get

$$FH_O^q = \frac{1}{1-q} \left\{ 1 - \frac{\sum_{l=0}^{L-1} [S(l)]^q (Up[l] + Rp[l])}{\left\{ \sum_{l=0}^{L-1} S(l) (U[l] + R[l]) \right\}^q} \right\}. \quad (24)$$

Therefore, the computation of FH_O^q for each pair of fuzzy parameters (a, c) is sped up from $O(L^2)$ to $O(L)$.

4.3. Thresholding Based on 2-D FTE and SHLA. To obtain the optimal threshold it is required to find the optimal combination of the fuzzy parameters. Therefore, the segmentation problem can be formulated as an optimization problem. SHLA [14] is an excellent global optimization algorithm proposed recently. It combines the benefits of the genetic-based memetic algorithm and the social behavior-based particle swarm optimization algorithms. In SFLA, the population consists of a set of frogs (solutions) that is partitioned into subsets referred to as memplexes. The different memplexes are considered as different cultures of frogs, each performing a local search. Within each memplex, the individual frogs hold ideas that can be influenced by the ideas of other frogs and evolve through a process of memetic evolution. After a defined number of memetic evolution steps, ideas are passed among memplexes in a shuffling process. The local search and the shuffling processes continue until defined convergence criteria are satisfied. Due to its rather fast convergence speed compared with other algorithm [15], SHLA is used to select the optimal fuzzy parameters in this paper. The major steps consist of the following.

Step 1. Calculate the 2-D histogram of the input image by (9) and produce four accumulating arrays by (19) and (20).

Step 2. Search the optimal fuzzy parameters (a, c) by SHLA, in which the fitness function is calculated according to (15).

Step 3. Set the threshold to be the crossover point of the membership function which has membership 0.5 implying the largest fuzziness [12], and segment the input image.

5. Experiments and Discussions

In order to verify the validity of the presented method, it is tested on a variety of synthetic and real images. To make purposeful yet effective comparisons, the selected thresholding methods for comparison are all entropy-based thresholding methods. They are Tao's 1D fuzzy Shannon entropy method [11], Sahoo's 2-D nonfuzzy Tsallis entropy method [8], and Cheng's 2-D fuzzy Shannon entropy method [12]. Additionally, the parameter q in (15) is set to be 0.7 as Sahoo did in [8], and the parameters of SHLA are set to be the values recommended in [14].

In order to evaluate the performance of the competing methods objectively, we conduct quantitative analysis on the results. There is no universally accepted objective criterion to evaluate the performance of segmentation methods. In this paper, our primary interest is the integrality and accuracy of object segmentation. Therefore, as Tao et al. did in [11], we also use the absolute error ratio as the comparison criterion. The absolute error ratio is defined as the ratio between the absolute error n_{diff} and the total number of pixels, N , of an image; that is,

$$r_{\text{err}} = \frac{n_{\text{diff}}}{N} \times 100\%, \quad (25)$$

TABLE 1: Thresholds and r_{err} of different methods in Figure 2(c).

Method	Tao	Sahoo	Cheng	Proposed
Threshold	77	121	120	116
r_{err}	14.34%	1.26%	0.99%	0.55%

where the absolute error n_{diff} is defined as the absolute difference in the number of object pixels between the ground truth image and the thresholding image obtained by each method [11].

5.1. Experiments on Synthetic Image. The image in Figure 2(a) is a simple synthetic image with 240×240 pixels and shows an English character "H" (with gray level 80) on grey background (with gray level 160). Ideal segmentation is shown in Figure 2(b). Figure 2(c) is the image into which Gaussian noises with zero mean and 0.005 variance are added, and the histogram of Figure 2(c) is shown in Figure 2(d).

The results obtained by the competing methods corresponding to Figure 2(c) are shown in Figure 3. It can be found that, as a 1-D method, Tao's method obtains the worst result; however the proposed method obtains the best one. Table 1 shows the thresholds and absolute errors obtained by competing methods. From Table 1, we can find that the absolute error ratio of the proposed method is lower than those of other methods.

5.2. Experiments on Real Images. In order to further verify the validity of the presented method, we tested it on a variety of real images including those selected from IEEE OT CBSV WS Series Bench [16] and those captured by ourselves using a Guide IR928 thermal camera. According to the method used in [11, 17], the average of the segmentation results got by five people familiar with thermal imagery is used as ground truth images here. The original images are shown in Figure 4, where (a)~(d) are selected from the public dataset and (e), (f) are captured by ourselves. Those images are captured in different seasons under different weather conditions and light strengths. It can be observed that the human targets in (a)~(c) are smaller while those in (d)~(f) are bigger. The corresponding ground truth images are shown in Figure 5. The results of the four competing methods are shown in Figures 6-9. The thresholds obtained by different methods are shown in Table 2, and absolute error ratios of the four competing methods are shown in Table 3.

From Figures 6-9, it can be found out that for image (a) all the four competing methods get rather good segmentation results by both visual inspection and quantitative analysis. However, for image (b), both Sahoo's method and Cheng's method fail. For image (c), Sahoo's method fails again, and Tao's method classifies nearly all the pixels as background. For image (d), only the proposed method segments out the infrared human targets successfully. For images (e) and (f), the results of Tao's method express some kind of over-segmentation. For all the six images, the results of the proposed method are rather reasonable.

TABLE 2: Thresholds obtained by different methods on images in Figure 4.

	Figure 4(a)	Figure 4(b)	Figure 4(c)	Figure 4(d)	Figure 4(e)	Figure 4(f)
Tao	170	153	207	76	189	181
Sahoo	153	66	58	65	103	130
Cheng	151	44	116	77	126	144
Proposed	167	128	102	128	122	128

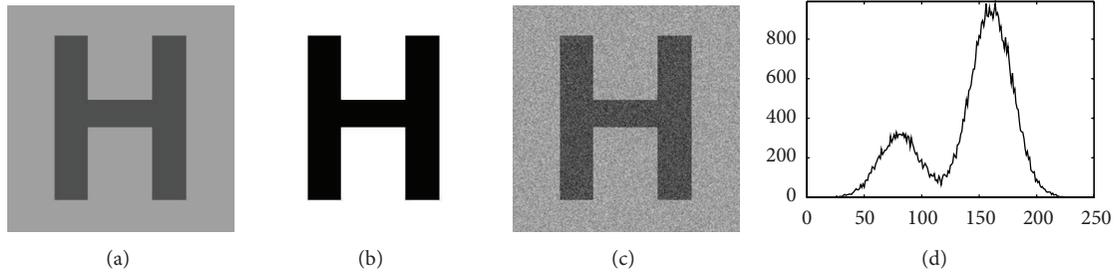


FIGURE 2: Synthetic images: (a) original image, (b) ground truth image, (c) noisy image, and (d) histogram of the noisy image.

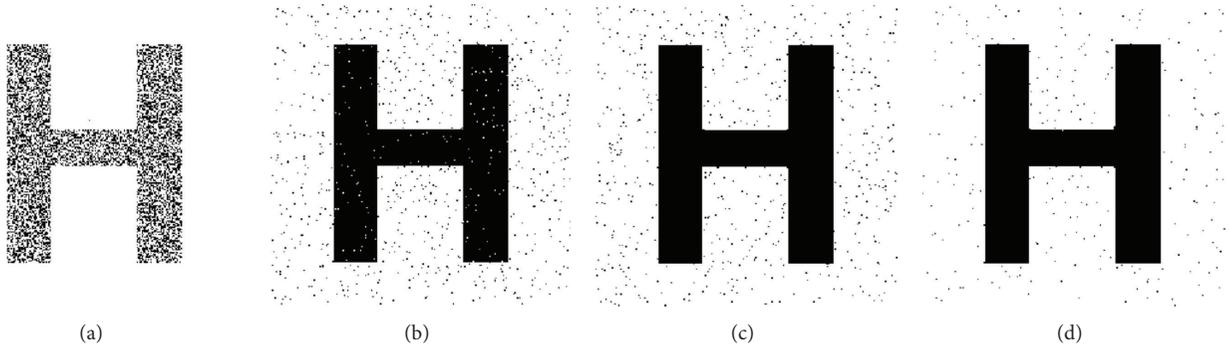


FIGURE 3: Thresholding results of noisy image obtained by different methods: (a) Tao's result, (b) Sahoo's result, (c) Cheng's result, and (d) our result.



FIGURE 4: Original images.

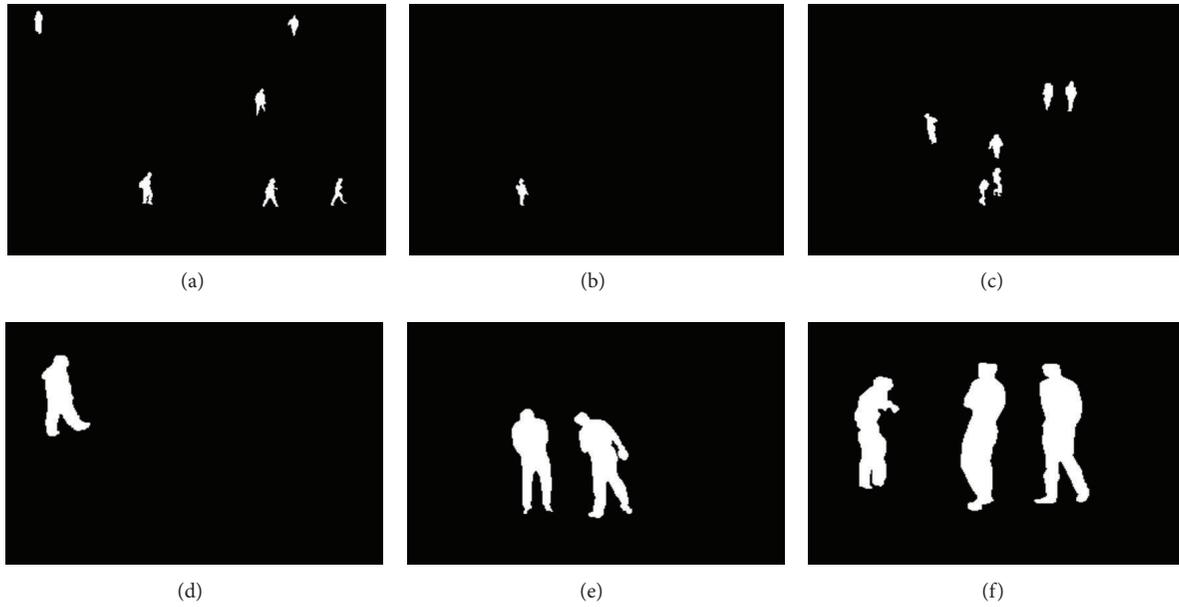


FIGURE 5: Ground truth images.

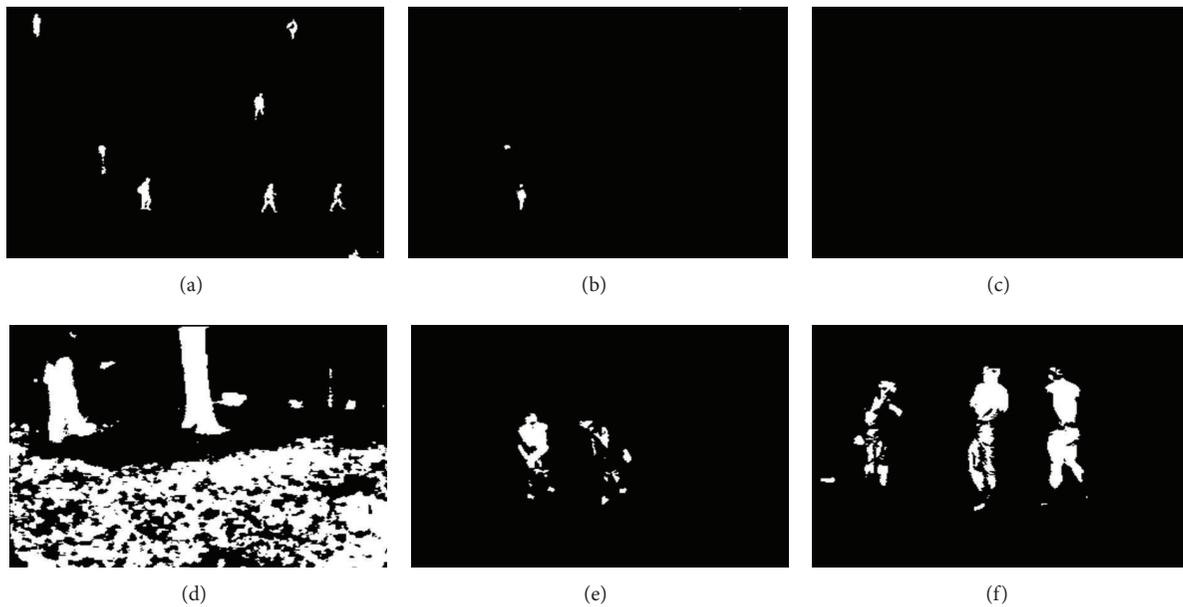


FIGURE 6: The results of Tao's method.

As for the analysis of time complexity, Sahoo's method used a purely exhaustive algorithm whose time complexity is $O(L^4)$. Cheng's method calculated the 2-D fuzzy entropy by exhaustive algorithm and then selected the fuzzy parameter sets by genetic algorithm, so the time complexity of Cheng's method is $O(G * P * L^2)$, in which G denotes the number of iterative generation and P denotes the size of population. Tao's method is a 1-D method in which ant colony optimization is adopted, and thus its time complexity is $O(G * P * L)$. By reducing the time complexity of the computation of 2D FTE

from $O(L^2)$ to $O(L)$ and employing SHLA to search for the optimal parameters of fuzzy membership function, the time complexity of our method is also $O(G * P * L)$.

When the experiments are carried out in Matlab (R2013a) on a computer with Intel Core i7-2600 CPU@2.93 GHz and 16 GB RAM, the running times (second) of different methods are listed in Table 4. It can be found from Table 4 that Sahoo's method takes the most time among all the methods, and the running time of our method is nearly equal to that of Tao's 1-D method.

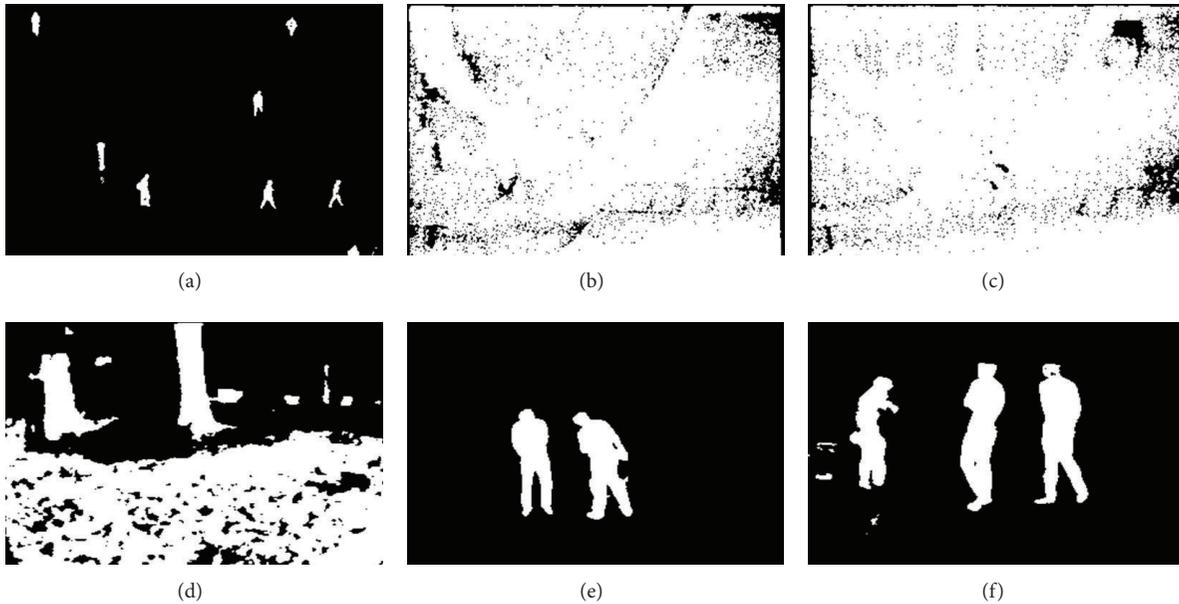


FIGURE 7: The results of Sahoo's method.

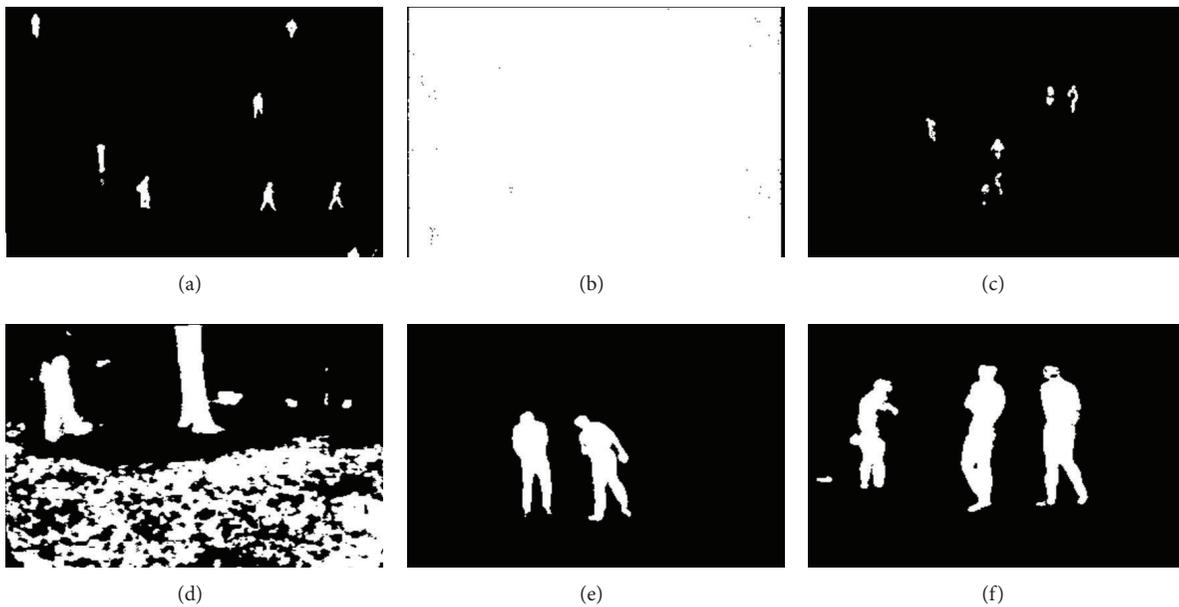


FIGURE 8: The results of Cheng's method.

6. Conclusion

In this paper, a thresholding method of infrared human image is developed based on 2-D Tsallis entropy and fuzzy sets theory. In the presented method, both the fuzziness of image and the spatial information of 2-D histogram are considered. To relieve the huge computation burden brought about by extending 1-D method to 2-D one, a fast algorithm is designed to reduce the time complexity of the computation of 2-D FTE from $O(L^2)$ to $O(L)$. And then SHLA is employed to speed

up the search of optimal threshold further. The experiments on both synthetic and real images show that the presented method is robust and effective.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

TABLE 3: r_{err} of different methods when segmenting images in Figure 4.

	Figure 4(a)	Figure 4(b)	Figure 4(c)	Figure 4(d)	Figure 4(e)	Figure 4(f)
Tao	0.30%	0.07%	1.13%	35.22%	4.09%	5.81%
Sahoo	0.36%	90.44%	90.99%	47.64%	0.68%	0.66%
Cheng	0.37%	97.99%	0.66%	33.88%	0.23%	1.44%
Proposed	0.29%	0.06%	0.65%	0.28%	0.07%	0.65

TABLE 4: Computation time of different methods for test images in Figure 4 (Second).

	Figure 4(a)	Figure 4(b)	Figure 4(c)	Figure 4(d)	Figure 4(e)
Tao	0.05282	0.03961	0.04594	0.05091	0.05371
Sahoo	186.32	153.47	156.37	162.63	145.67
Cheng	1.3048	1.0358	1.2547	1.09641	1.0846
Proposed	0.05094	0.04259	0.04807	0.04986	0.05237

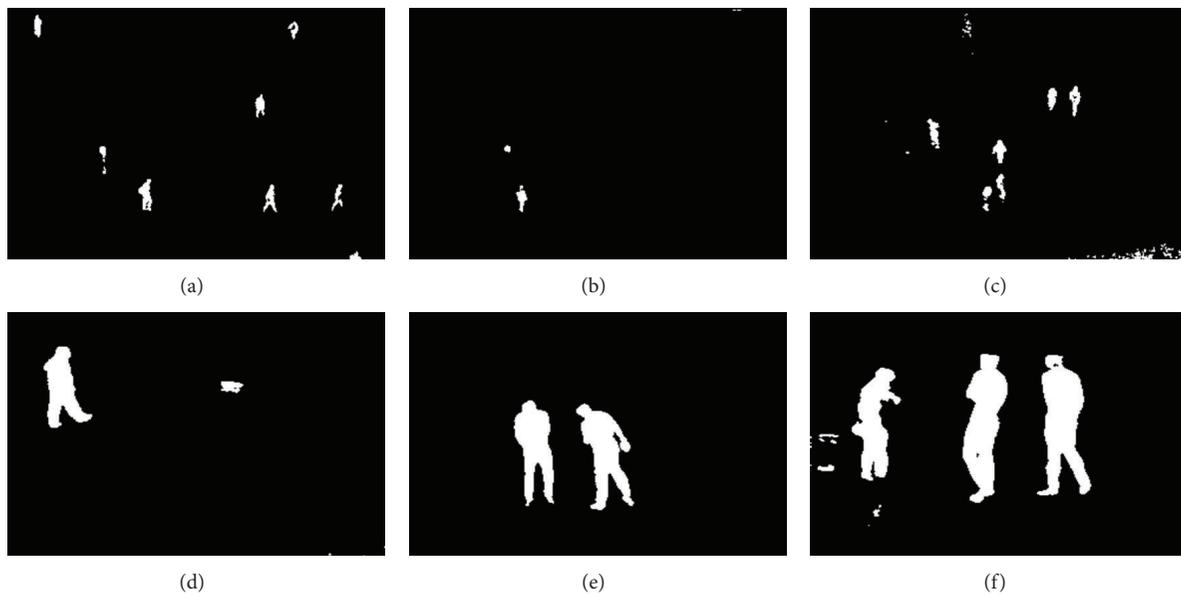


FIGURE 9: The results of our method.

Acknowledgments

This work is supported by National Natural Science Foundation of China (no. 61302178), Natural Science Foundation of Guangxi Zhuang Autonomous Region (no. 2013GXNSFAA019347 and no. 2013GXNSFAA019336), Research Foundation of Education Bureau of Guangxi Zhuang Autonomous Region (no. 2012LX659 and No. 2013LX088) and Natural Science Foundation of Guangxi University of Science and Technology (no. 1307111).

References

- [1] P.-Y. Yin, "Maximum entropy-based optimal threshold selection using deterministic reinforcement learning with controlled randomization," *Signal Processing*, vol. 82, no. 7, pp. 993–1006, 2002.
- [2] Z. Li, C. Liu, G. Liu, X. Yang, and Y. Cheng, "Statistical thresholding method for infrared images," *Pattern Analysis and Applications*, vol. 14, no. 2, pp. 109–126, 2011.
- [3] F. Nie, C. Gao, Y. Guo, and M. Gan, "Two-dimensional minimum local cross-entropy thresholding based on co-occurrence matrix," *Computers and Electrical Engineering*, vol. 37, no. 5, pp. 757–767, 2011.
- [4] J. F. Li, W. G. Gong, W. H. Li, and X. Liu, "Robust pedestrian detection in thermal infrared imagery using the wavelet transform," *Infrared Physics and Technology*, vol. 53, no. 4, pp. 267–273, 2010.
- [5] F. Nie, C. Gao, and Y. Guo, "Infrared human image segmentation using fuzzy Havrda-Charv't entropy and chaos PSO algorithm," *Journal of Computer-Aided Design and Computer Graphics*, vol. 22, no. 1, pp. 129–135, 2010.
- [6] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–168, 2004.

- [7] M. Portes de Albuquerque, I. A. Esquef, A. R. Gesualdi Mello, and M. Portes de Albuquerque, "Image thresholding using Tsallis entropy," *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1059–1065, 2004.
- [8] P. K. Sahoo and G. Arora, "Image thresholding using two-dimensional Tsallis-Havrda-Charvát entropy," *Pattern Recognition Letters*, vol. 27, no. 6, pp. 520–528, 2006.
- [9] A. De Luca and S. Termini, "A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory," *Information and Control*, vol. 20, no. 4, pp. 301–312, 1972.
- [10] H. D. Cheng, C. H. Chen, H. H. Chiu, and H. Xu, "Fuzzy homogeneity approach to multilevel thresholding," *IEEE Transactions on Image Processing*, vol. 7, no. 7, pp. 1084–1088, 1998.
- [11] W. Tao, H. Jin, and L. Liu, "Object segmentation using ant colony optimization algorithm and fuzzy entropy," *Pattern Recognition Letters*, vol. 28, no. 7, pp. 788–796, 2007.
- [12] H. D. Cheng, Y. H. Chen, and X. H. Jiang, "Thresholding using two-dimensional histogram and fuzzy entropy principle," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 732–735, 2000.
- [13] M. Zhao, A. M. N. Fu, and H. Yan, "A technique of three-level thresholding based on probability partition and fuzzy 3-partition," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 3, pp. 469–479, 2001.
- [14] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.
- [15] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005.
- [16] J. W. Davis and M. A. Keck, "A two-stage template approach to person detection in thermal imagery," in *Proceedings of the 7th IEEE Workshop on Applications of Computer Vision (WACV '05)*, pp. 364–369, January 2005.
- [17] V. Sharma and J. W. Davis, "Simultaneous detection and segmentation of pedestrians using top-down and bottom-up processing," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 3666–3673, June 2007.

Research Article

Image Denoising Using Total Variation Model Guided by Steerable Filter

Wenxue Zhang,¹ Yongzhen Cao,¹ Rongxin Zhang,¹ and Yuanquan Wang²

¹ Radiation Oncology Department, Tianjin Medical University General Hospital, Tianjin 300054, China

² School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin 300094, China

Correspondence should be addressed to Yuanquan Wang; yqwang@bit.edu.cn

Received 7 October 2013; Accepted 8 December 2013; Published 5 January 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Wenxue Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an adaptive total variation (TV) model by introducing the steerable filter into the TV-based diffusion process for image filtering. The local energy measured by the steerable filter can effectively characterize the object edges and ramp regions and guide the TV-based diffusion process so that the new model behaves like the TV model at edges and leads to linear diffusion in flat and ramp regions. This way, the proposed model can provide a better image processing tool which enables noise removal, edge-preserving, and staircase suppression.

1. Introduction

Image denoising is a fundamental task in the community of image processing, but there is always a dilemma for the denoising algorithms to simultaneously remove noise and to preserve edges. For example, the Gaussian filter can smooth noise effectively, but it also blurs the edges since it is just a low-pass filter which cannot discern noise and edges. Recent advances on this topic have brought considerable improvement in denoising performance, such as the kernel regression (LARK) [1], bilateral filter [2], patch-based method [3, 4], BM3D [5], and the partial differential equation (PDE) based methods [6]. In [7], the author presented a tutorial on these state-of-the-art denoising methods, and it has been shown that the LARK takes the bilateral filter and the nonlocal mean (NLM) [3] as special cases and is closely related to the anisotropic diffusion [8]. In this work, we intend to concentrate on the PDE-based methods.

During the past two decades, the diffusion partial differential equations (PDEs) which can be considered as stemming from the heat diffusion process have gained popularity in the community of image processing, with a particular emphasis on image denoising [6]. One typical algorithm is the anisotropic diffusion proposed by Perona and Malik (PM model) [8], which is able to reduce diffusion amount around boundaries and achieve a good trade-off between

noise removal and edge preservation. In [9], the authors presented an improvement of the PM model using the difference eigenvalue. In [10], anisotropic second- and fourth-order diffusions are coined based on the gradient vector convolution. In [11], the second-order diffusion is combined with the fourth-order one for image restoration, and, in [12], the convolutional virtual electric field is introduced into the diffusion model. The directional Laplacian is also introduced into the diffusion equation for image restoration [13]. The total variation model is another typical model which minimizes the following functional [14]:

$$E_{TV} = \int_{\Omega} \left(|\nabla u| + \frac{1}{2} \lambda (u - u_0)^2 \right) dx dy, \quad (1)$$

where Ω is the image domain, ∇ is the gradient operator, and $\lambda > 0$ is a weight parameter. $u(x, y)$ is the unknown image to be recovered and $u_0(x, y)$ is the input noisy image. The norm $\int_{\Omega} |\nabla u|$ is the regularization term and the norm $\int_{\Omega} (u - u_0)^2$ is the fidelity term. The regularization term measures the amount of oscillation found in the function $u(x, y)$ and allows for discontinuities while disfavours oscillations; as a result, the TV model performs very well on preserving edges and removing noise.

Since its debut in 1992, there has been a flurry of works devoted to the TV model, such as the color TV [15], total

generalized variation [16], Bregman iteration [17], L1-norm data fidelity [18], edge oriented TV [19], and the theoretical analysis [20], among others. It is interesting that a MTV model is introduced in [21]. However, the TV model, as well as the Perona-Malik anisotropic diffusion model [8], suffers from the so-called staircasing effect; that is, it transforms the ramp regions into piecewise constant regions. These staircases are visually unpleasant and can be falsely recognized as edges. A natural way to overcome this shortcoming is to increase the order of derivatives in the diffusion model, and the fourth-order PDEs have been of special interest in recent years [22–24]. Although the fourth-order PDEs are free of staircasing effect, there is still staircasing effect in the second-order PDE models themselves [8, 14]. Another typical improvement of the TV model is proposed using a variable exponent $p(x)$ in the TV energy by Blomgren et al. (BCMW) [25] as follows:

$$E_{\text{BCMW}} = \int_{\Omega} \left(|\nabla u|^{p(|\nabla u|)} + \frac{1}{2} \lambda (u - u_0)^2 \right) dx dy, \quad (2)$$

where $\lim_{g \rightarrow 0} p(g) = 2$ and $\lim_{g \rightarrow \infty} p(g) = 1$ with $p(x)$ monotonically decreasing. This is a nonconvex minimization problem [26]; it behaves like TV model at edges ($|\nabla u| \rightarrow \infty$) and like linear heat diffusion in flat regions ($|\nabla u| \rightarrow 0$). After this model was proposed, there have been several works focused on this model by proposing different $p(x)$ functions; for example, the authors in [27, 28] proposed to employ piecewise functions for $p(x)$; the difficulties of these works are that how to set the thresholds in the piecewise functions. In [27], the author also suggested to use a smoothed version of the noisy image in $p(|\nabla u|)$, that is, $p(|\nabla G_{\sigma} * u_0|)$. This choice makes the minimization problem convex [26]; however, the gradient magnitude $|\nabla G_{\sigma} * u_0|$ can not characterize the edge and ramp region very well. Very recently, Chen et al. proposed an ATV model by designing an exponent function $p(x)$ using the difference curvature (i.e., second-order directional derivative difference) [29]; $D = \|\|u_{\eta\eta}\| - \|u_{\xi\xi}\|$, $u_{\eta\eta}$ and $u_{\xi\xi}$ are the second-order derivatives along and parallel to the gradient direction ∇u , respectively. However, the variant D is sensitive to noise owing to the second-order derivatives. What is more, it is difficult to control the diffusion amount since the D is time varying. There are also other works concerning the staircase reduction including the Gauss curvature driven diffusion [30], the linear regression method in the neighborhood filters [31], and the modified structure-based diffusion [32].

In this paper, our purpose also aims at devising a more effective exponent $p(x)$ for the BCMW model in (2). Motivated by the fact that the steerable filter proposed in [33] performs very well in orientation analysis and edge detection [33, 34], the local energy measured by the steerable filter is employed to characterize the edges, ramps, and flat regions. Based on this local energy, we devise an exponent $p(x)$ continuously changing from 1 to 2 to regularize the TV diffusion and also devise an adaptive weight for the fidelity term; this corresponding model is referred to as STV (steerable filter guided TV) model. The proposed STV model is mathematically convex and behaves like the TV model at edges and like the heat diffusion in flat regions. Comparative

results on both synthetic and natural images demonstrate that this STV model not only has a good performance on avoiding the staircases but also performs well on preserving the edges.

The remainder of this paper is organized as follows. In Section 2, the steerable filters are introduced in brief and the local energy measured by the steerable filter is presented. In Section 3, the STV model is presented based on the local energy. In Section 4, the performance of the STV model is demonstrated by experiment and comparison. Finally, conclusions are drawn in Section 5.

2. Steerable Filter and Local Energy

Oriented filters are useful in many early vision and image processing tasks, such as texture analysis and edge detection [35–37]. In [33, 34], Freeman and Adelson proposed the framework named steerable filter to create an oriented filter by describing it as a linear combination of basis filters. Under this framework, they designed the steerable quadrature pairs using the n th derivative of a Gaussian and its Hilbert transform. This steerable quadrature filter measures local energy and, based on the local energy, methods have been devised to analyze orientation and study structures like contours and edges [33, 34]. Here we employ the local energy to indicate edges and flat regions and to guide the diffusion of the TV model. In the following, we describe the steerable quadrature filter which follows the description in [33].

Generally, the steerable filter can be written in the following form:

$$f^{\theta}(x, y) = \sum_{j=1}^M k_j(\theta) f^{\theta_j}(x, y), \quad (3)$$

where $k_j(\theta)$ is the interpolation function, $f^{\theta_j}(x, y)$ is the basis function, and M is the number of the basis functions. This expression tells us that the steerable filter $f(x, y)$ can be written as a linear combination of M rotated versions of itself $f^{\theta_j}(x, y)$. The steering theorems in [33, 34] answer the three questions: what functions $f(x, y)$ can satisfy (3), how many terms, M , are required in the sum, and what the interpolation functions, $k_j(\theta)$, are.

Under the steerable filter framework, the n th derivative of a Gaussian G_n^{θ} at arbitrary orientation θ using x - y separable basis filters is presented in [33], and its Hilbert transform H_n^{θ} is also approximated by finding the least square fit to a polynomial times a Gaussian. G_n^{θ} and H_n^{θ} form a steerable quadrature pair, which allow for analyzing spectral strength independent of phase and allow for synthesizing filters of a given frequency response with arbitrary phase [34]. In this paper, the second derivative G_2^{θ} and its Hilbert transform H_2^{θ} are employed to measure local energy. The G_2^{θ} takes the following form:

$$G_2^{\theta} = k_a(\theta) G_{2a} + k_b(\theta) G_{2b} + k_c(\theta) G_{2c}, \quad (4)$$

where $k_a(\theta) = \cos^2(\theta)$, $G_{2a} = 0.9213(2x^2 - 1) \exp(-(x^2 + y^2))$, $k_b(\theta) = -2 \cos(\theta) \sin(\theta)$, $G_{2b} = 1.843xy \exp(-(x^2 +$

y^2), $k_c(\theta) = \sin^2(\theta)$, and $G_{2c} = 0.9213(2y^2 - 1) \exp(-(x^2 + y^2))$. Similarly, the H_2^θ reads

$$H_2^\theta = l_a(\theta) H_{2a} + l_b(\theta) H_{2b} + l_c(\theta) H_{2c} + l_d(\theta) H_{2d}, \quad (5)$$

where $l_a(\theta) = \cos^3(\theta)$, $H_{2a} = 0.9780(-2.254x + x^3) \exp(-(x^2 + y^2))$, $l_b(\theta) = -3 \cos^2(\theta) \sin(\theta)$, $H_{2b} = 0.9780(-0.7515 + x^2)(y) \exp(-(x^2 + y^2))$, $l_c(\theta) = 3 \cos(\theta) \sin^2(\theta)$, $l_d(\theta) = -\sin^3(\theta)$, $H_{2c} = 0.9780(-0.7515 + y^2)(x) \exp(-(x^2 + y^2))$, and $H_{2d} = 0.9780(-2.254y + y^3) \exp(-(x^2 + y^2))$. Therefore, the local energy $E_2(\theta)$ is measured by

$$E_2(\theta) = [G_2^\theta]^2 + [H_2^\theta]^2. \quad (6)$$

Here, we are more interested in the maximal response of $E_2(\theta)$ at certain orientation θ_d :

$$s(\theta_d) = \max_{0^\circ \leq \theta \leq 360^\circ} (E_2(\theta)). \quad (7)$$

It was shown that the measured orientation and response accurately reflect the oriented structures of the input image [33, 34]; consequently, $s(\theta_d)$ is employed to devise an effective exponent $p(x)$ for the BCMW model in (2).

In previous works, intensity gradient [27] and difference curvature [29] are employed as the indicator to discriminate edges and ramps. Comparison between our new indicator $s(\theta_d)$ and those is shown in Figure 1. From Figure 1, we can observe that the difference curvature is sensitive to the structures in image, including the object boundaries and the false edges caused by inhomogeneity and noises. The difference curvature in [29] is calculated from the being denoised image in each iteration; this makes the difference curvature less sensitive to noise, and, however, it also makes the ATV model difficult to control the diffusion amount. The gradient also captures the object boundaries and is less sensitive to the false edges stemming from inhomogeneity and noise than the difference curvature; however, it is calculated in advance and the false edges also affect the denoising effect. Meanwhile, the local energy from steerable filter is immune to inhomogeneity and robust to noise and captures the boundaries accurately. Although the local energy is not single-pixel wide, the max response characterizes the step edges very well (see the red line in Figure 1(a)), and the values nearby the max response are very small. Thus, the new indicator $s(\theta_d)$ is a good choice to regularize the TV model for image denoising.

3. STV Model

Based on the new indicator $s(\theta_d)$ derived from the steerable filter in (7), we devise a new exponent function $p(x)$ and an adaptive weight $\lambda(x)$ and propose the STV model as follows:

$$E_{\text{STV}} = \int_{\Omega} \left(|\nabla u|^{p(s)} + \frac{1}{2} \lambda(s) (u - u_0)^2 \right) dx dy, \quad (8)$$

where the functions $p(s)$ and $\lambda(s)$ are

$$p(s) = \frac{2.0}{1.0 + \sqrt{s}} \quad (9)$$

$$\lambda(s) = k \cdot \sqrt{s},$$

where \bar{s} is the normalized energy $\bar{s} = (s - s_{\min}) / (s_{\max} - s_{\min})$, s_{\max} and s_{\min} are the maximal and minimal values of s throughout the entire image, respectively, and k is a positive constant parameter. Since $\bar{s} \in [0, 1]$, the function $p(s)$ monotonically decreases from 2 to 1, in flat and ramp regions, $p(s) \rightarrow 2$, $\lambda(s) \rightarrow 0$, at sharp edges, $p(s) \rightarrow 1$, $\lambda(s) \rightarrow k$. Thus, the STV model approximates the TV model with the weight of the fidelity term k near edges and approaches the linear heat flow away from the edges. Since the STV model can adapt the exponent function $p(s)$ and the weight of the fidelity term $\lambda(s)$ in accordance with the image feature, it can preserve edges and fine details and prevent the staircasing effect simultaneously.

Let $g(x) = x^{p(s)}$, $x \geq 0$, $p(s) \in [1, 2]$; then $g'(x) > 0$ and $g''(x) > 0$, $x \geq 0$; that is, $g(x)$ is an increasing convex function, and the problem (8) has a unique solution. Using the calculus of variation, we can obtain the solution of (8) by solving the following Euler equation:

$$\frac{\partial u}{\partial t} = \text{div}(c(|\nabla u|) \nabla u) - \lambda(s)(u - u_0), \quad (10)$$

where $c(x) = g'(x)/x$. According to the orthogonal decomposition in [38], (10) can be further written in the following form:

$$\frac{\partial u}{\partial t} = p(s)(p(s) - 1) |\nabla u|^{p(s)-2} u_{\eta\eta} + p(s) |\nabla u|^{p(s)-2} u_{\xi\xi} - \lambda(s)(u - u_0), \quad (11)$$

where $u_{\eta\eta} = (u_{xx}u_x^2 + 2u_xu_yu_{xy} + u_{yy}u_y^2)/(u_x^2 + u_y^2)$ and $u_{\xi\xi} = (u_{xx}u_y^2 - 2u_xu_yu_{xy} + u_{yy}u_x^2)/(u_x^2 + u_y^2)$ represent the second derivatives in the normal and tangent directions of the level sets of function $u(x, y)$, respectively. Therefore, the first and the second terms denote the diffusion in the normal and tangent directions of the level sets of function $u(x, y)$ with different diffusion coefficients. Let $c_\eta = p(s)(p(s) - 1) |\nabla u|^{p(s)-2}$, $c_\xi = p(s) |\nabla u|^{p(s)-2}$; the diffusion ratio in the two directions is

$$R = \frac{c_\eta}{c_\xi} = \frac{p(s)(p(s) - 1) |\nabla u|^{p(s)-2}}{p(s) |\nabla u|^{p(s)-2}} = p(s) - 1. \quad (12)$$

It can be seen from this diffusion ratio that, when the image gradient is small, $p(s)$ approaches 2 and $R \approx 1$; that is, the diffusion is isotropic. This property implies that the STV model can smooth the image and prevent staircases in flat and ramp regions. When the image gradient is high, $p(s)$ approaches 1 and $R \approx 0$; that is, the diffusion is mainly in the tangent direction; thus, the STV model can smooth the image and preserve edges.

Finite difference method is employed to solve (11); the forward difference is used for the time derivative and central

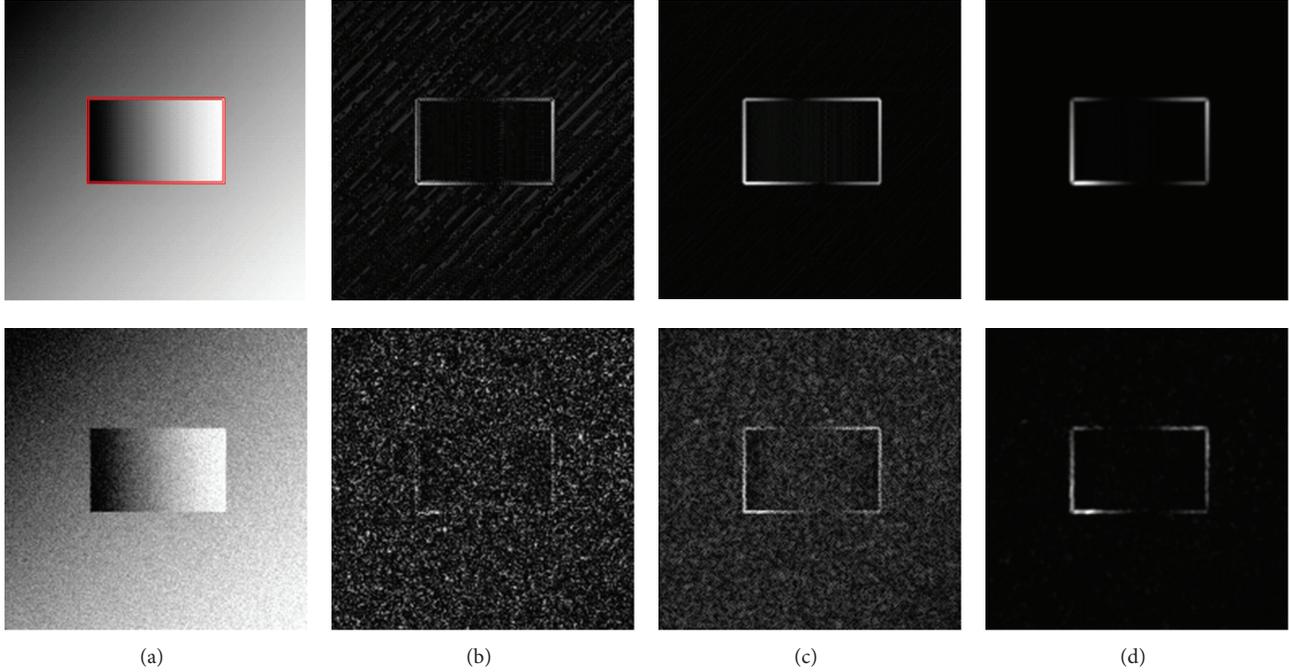


FIGURE 1: Noise-free (top row) and noisy (bottom row) images with ramps. (a) Original images, (b) difference curvature, (c) intensity gradient, and (d) local energy derived from steerable filter.

differences are used for spatial derivatives. These difference operators are given by

$$\begin{aligned}
 \frac{\partial u}{\partial t} &= \frac{u^{n+1} - u^n}{\Delta t}, & u_x &= \frac{(u_{i-1,j} - u_{i+1,j})}{2}, \\
 u_y &= \frac{(u_{i,j-1} - u_{i,j+1})}{2}, & u_{xx} &= u_{i-1,j} + u_{i+1,j} - 2u_{i,j}, \\
 & & u_{yy} &= u_{i,j-1} + u_{i,j+1} - 2u_{i,j}, \\
 u_{xy} &= \frac{(u_{i-1,j-1} + u_{i+1,j+1})}{4} - \frac{(u_{i-1,j+1} + u_{i+1,j-1})}{4}.
 \end{aligned} \tag{13}$$

Thus, the discrete approximation of the directional derivatives of $u(x, y)$, that is, $u_{\eta\eta}$ and $u_{\xi\xi}$, is simply derived after direct substitution of the difference operators listed above into $u_{\eta\eta}$ and $u_{\xi\xi}$. For the measurement of convergence, the “normalized step difference energy” (NSDE) [30] was calculated at each iteration. It is defined as $NSDE = |u_k - u_{k-1}|^2 / (|u_k|^2 \cdot M \cdot N)$, where u_k and u_{k-1} denote the image at k and $k - 1$ iteration, respectively. The image dimension is $M \cdot N$. In Figure 2, we demonstrated the effectiveness of the proposed STV model and its NSDE graph. It can be observed that the result is clear and free of staircase, and the edges are sharp. The NSDE graph also shows that the STV model converged efficiently. In fact, when the iteration reaches 25, the NSDE is less than 10^{-5} .

4. Experimental Results

In this section, we demonstrate the effectiveness of the proposed STV model by experiment and comparison on both synthetic and natural images. Nine well-known models are employed for comparison, including the TV model [14], the PM model [8], YK model [23], the ATV model [29], the DED model [9], the Hajiaboli’s fourth-order model (Hajiaboli) [24], the LARK [1], the BM3D [5], and the NLM [3]. The peak signal-to-noise ratio (PSNR) and the mean structure similarity (MSSIM) [39] are employed as object indexes to evaluate the image quality of the filtered images. The MSSIM ranges from 0 to 1 and if the filtered image is identical to the noise-free one, it is 1. We also visually evaluate the denoised images. The results of the NLM are yielded from the IPOL (http://www.ipol.im/pub/algo/bcm_non_local_means_denoising/) website, so the parameters for the NLM are default. We adopt the software packages of the LARK (<http://users.soe.ucsc.edu/~milanfar/software/>) and BM3D (http://www.cs.tut.fi/~foi/GCF-BM3D/index.html#ref_papers) and the parameters for both models are unchanged as in the package. The parameters for the other models are shown in Table 1. The time step and k (or λ) are adjusted to be as close as possible to those in other literatures such as [23]. It is well known that the iteration number is also critical to control the diffusion amount. We calculate the PSNR and MSSIM values at each iteration for the iteratively solved models, including the YK, PM, TV, ATV, STV, Hajiaboli, DED, and LARK models; the algorithms are terminated when the associated MSSIM values reach maximum. The runtime of each model for each experiment is also listed in Table 1; it is obvious that the BM3D is the fastest, and then

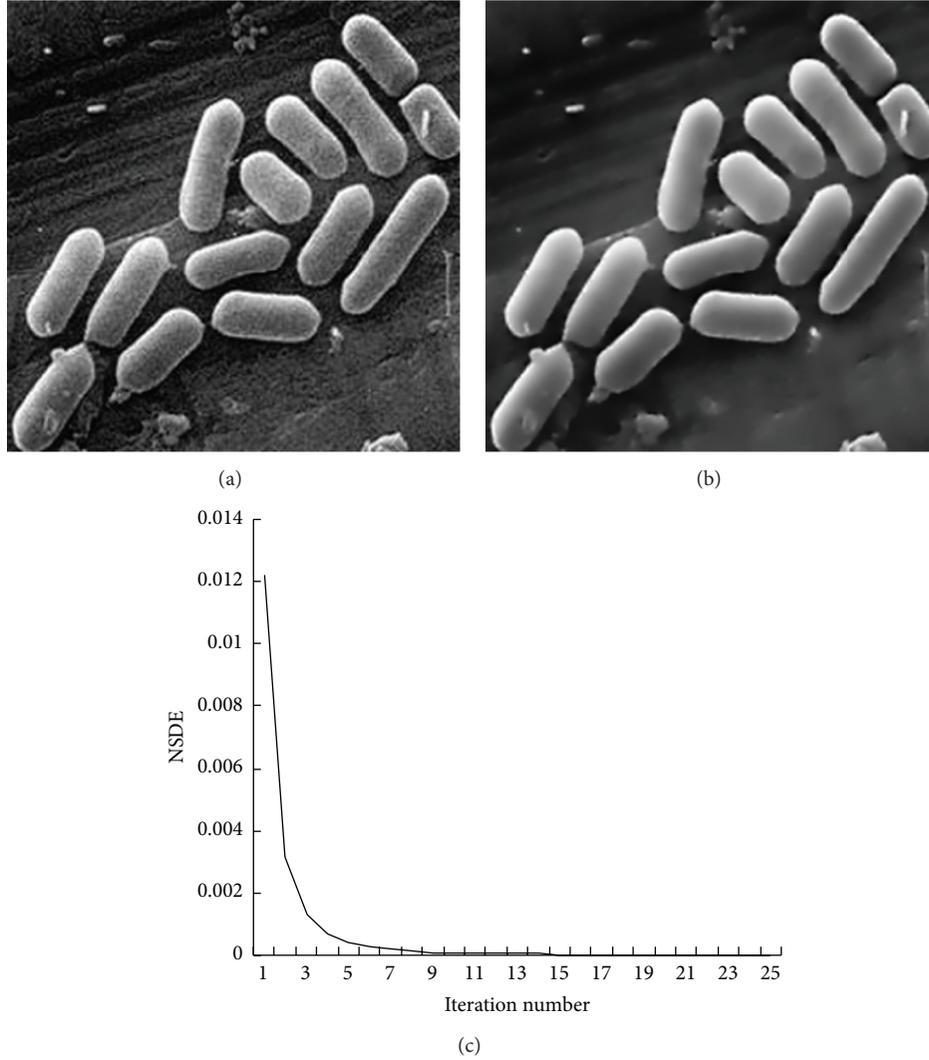


FIGURE 2: Demonstration of the STV model and its NSDE graph. (a) Noisy medical image, (b) denoised image, and (c) the mean NSDE graph.

TABLE 1: Parameters for the nine models.

Method	Time step	k/λ	Iteration number/time in second		
			Figure 3	Figure 4	Figure 5
You-Kaveh	0.2	3	474/81.2	514/1367.9	482/176.3
TV model	0.2	0.01	99/14.2	91/177.6	90/28.9
PM model	0.2	3	147/14.2	215/282.5	208/42.4
ATV	0.1	1	29/6.5	15/35.1	12/5.8
STV	0.1	1	81/12.4	28/53.1	24/8.1
DED	0.1	—	1876/241.0	3500/5230	4100/1118
Hajiaboli	0.03	7	570/30.8	160/175.3	160/29.4
LARK	Default as in the package		7/192.1	9/1844.2	9/473.1
BM3D	Default as in the package		2.4	13.5	3.5

the ATV and STV models; the LARK and DED models are the most time consuming.

The first experiment is a noisy synthetic image of ramps, which is shown in Figure 1(a). There are no complicated structure and texture in the image; however, the ramps

and the sharp edges should be preserved. Figure 3 shows the results of the ten models. Visually, the PM and BM3D models perform the best on preserving the sharp edges and, apart from the DED model, the YK and LARK perform the worst. However, there are serious staircases in the results by

TABLE 2: PSNR and MSSIM of Figure 3.

	Figures										
	Figure 3(a)	Figure 3(b)	Figure 3(c)	Figure 3(d)	Figure 3(e)	Figure 3(f)	Figure 3(g)	Figure 3(h)	Figure 3(i)	Figure 3(j)	Figure 3(k)
PSNR	25.73	35.65	37.28	37.28	36.67	37.47	35.76	37.14	38.38	37.57	34.13
MSSIM	0.348	0.903	0.925	0.922	0.924	0.927	0.866	0.925	0.929	0.926	0.856

TABLE 3: PSNR and MSSIM of Figure 4.

	Figures										
	Figure 4(a)	Figure 4(b)	Figure 4(c)	Figure 4(d)	Figure 4(e)	Figure 4(f)	Figure 4(g)	Figure 4(h)	Figure 4(i)	Figure 4(j)	Figure 4(k)
PSNR	22.15	29.87	31.32	29.13	30.43	31.46	31.57	32.61	32.89	31.15	28.53
MSSIM	0.346	0.792	0.842	0.783	0.828	0.855	0.829	0.869	0.866	0.835	0.810

TABLE 4: PSNR and MSSIM of Figure 5.

	Figures										
	Figure 5(a)	Figure 5(b)	Figure 5(c)	Figure 5(d)	Figure 5(e)	Figure 5(f)	Figure 5(g)	Figure 5(h)	Figure 5(i)	Figure 5(j)	Figure 5(k)
PSNR	22.22	28.43	29.96	28.05	28.77	30.18	30.33	30.75	31.66	29.77	27.23
MSSIM	0.428	0.826	0.879	0.820	0.863	0.890	0.854	0.893	0.892	0.882	0.790

the PM and TV models; there are also flow-like structure in the results by the LARK and BM3D models. The flow-like structure is more obviously in the Lena experiment for the LARK model; this will be shown in Figure 4. The result by the NLM model is somewhat foggy; see Figure 3(g). The result by the YK model is staircase-free, but there are dark spots; see Figure 3(b). The Hajiaboli model is also a fourth-order model and the result is much better than that of the YK model due to its anisotropic nature and comparable to that of the STV model; see Figure 3(j). The results by the STV and ATV models are visually comparable and staircase-free, but the PSNR and MSSIM values shown in Table 2 manifest that the STV model is superior to the ATV model. The result of the DED model is the worst from both visual inspection and PSNR and MSSIM indices; the reason is that there is no parameter playing a role similar to that of the parameter k in the PM model. From Table 2, we also see that the BM3D has the best performance, the STV and Hajiaboli model are the second best, and the NLM, YK, and the DED models are the last. Clearly, the STV performs the best among the diffusion based methods.

The popular Lena and Pepper images are employed as the second and third examples. The noisy images in Figures 3~5 are yielded from the IPOL website by specifying the noise deviation to 20. The results of the ten models on Lena image are listed in Figure 4 and those on the pepper image are in Figure 5, but only the face region of the Lena image is shown for the sake of clarity. From Figures 4 and 5, it is obvious that, for the YK model, the results are misty apart from the speckles; for the PM model, there are also speckles and staircases which are visually unpleasant. For the DED model, since there is no parameter playing the role of the parameter k in the PM model, the results are the worst from both visual inspection and PSNR and MSSIM indices. Similar results for the YK and PM models were also reported in

[23]. Although there is no speckle for the TV model, there are many staircases, particularly, in the face region. While for the ATV model, there is no staircase, but there are dark spots as if there are acne and pimple in Lena's face. We also observed acne and pimple in Lena's face in the result of the Hajiaboli model. The results of the NLM are also misty. The LARK model suffers from the flow-like structure in Lena's face, especially around the eyes and mouth. On the contrary, the proposed STV model yields clear and staircase-free result. The results of the BM3D model are also clear and staircase-free, except that there is slight flow-like structure around Lena's chin. The PSNR and MSSIM values of each model in Figures 4 and 5 are reported in Tables 3 and 4, respectively. From these two tables, one can see that, although the BM3D and LARK models yield better PSNR than the STV does, the associated MSSIM values are comparable. The STV model outperforms the other models in terms of the MSSIM, including the NLM method, and performs the best among the diffusion based methods.

5. Conclusion

In this paper, we have proposed a nonlinear diffusion process (STV) for image filtering. The proposed STV model is coined by introducing the local energy measured by the steerable filter into the total variation (TV) model. The main result is that the STV model behaves like the TV model at edges and like the linear heat diffusion in ramp and flat regions, so that the STV model can simultaneously remove noises, preserve edges, and suppress staircases. Experiments have been conducted on both synthetic and real images, and comparisons have been launched with the classical and state-of-the-art models such as the YK, PM, TV, ATV, Hajiaboli, DED, NLM, LARK, and BM3D models. We have evaluated

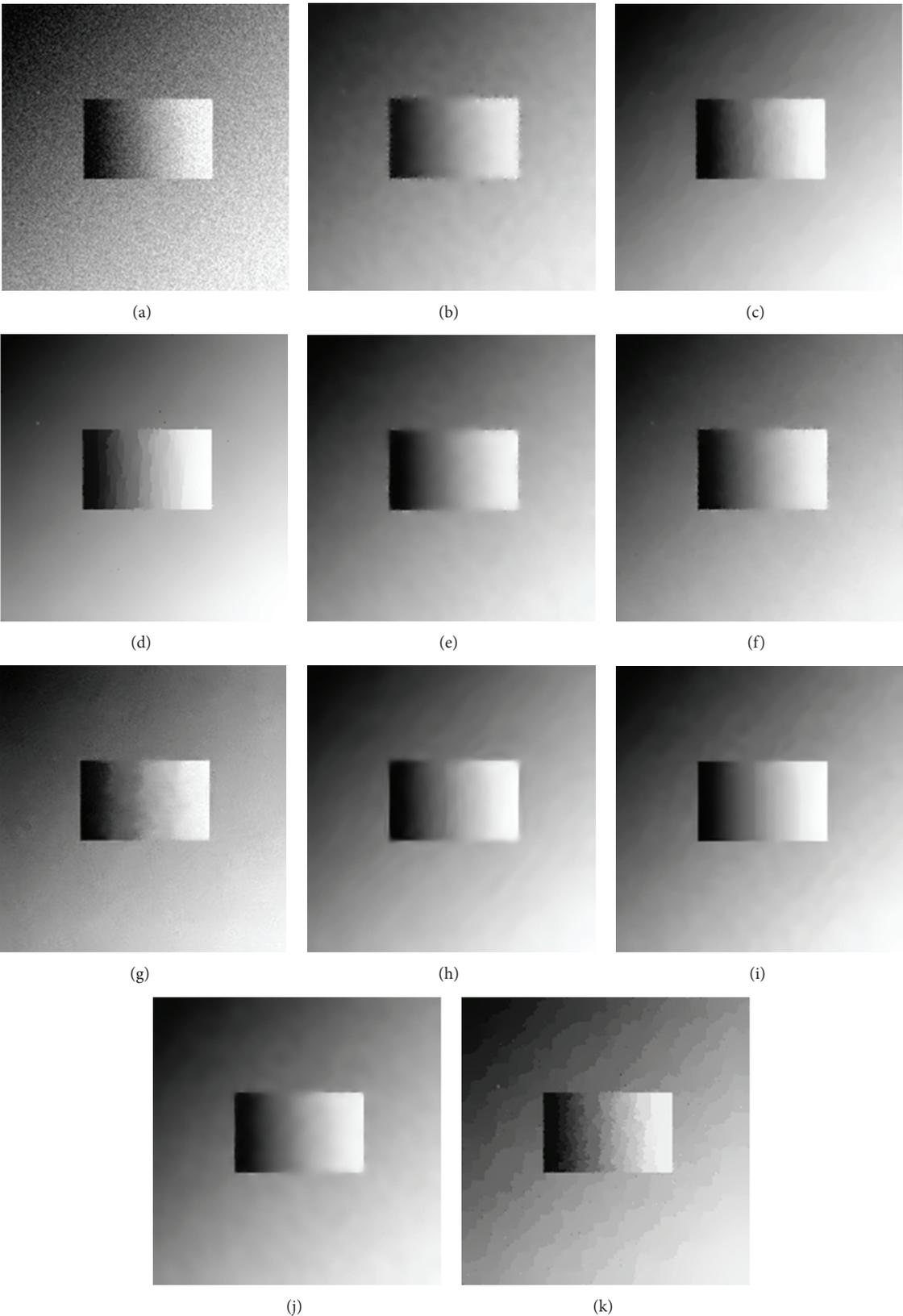


FIGURE 3: Demonstration and comparison of a synthetic image. (a) Noisy image, filtered images by (b) the You-Kaveh fourth-order model, (c) the TV model, (d) the PM model, (e) the ATV model, (f) the STV model, (g) the nonlocal mean, (h) the LARK model, (i) the BM3D model, (j) Hajjiboli's model, and (k) the DED model.

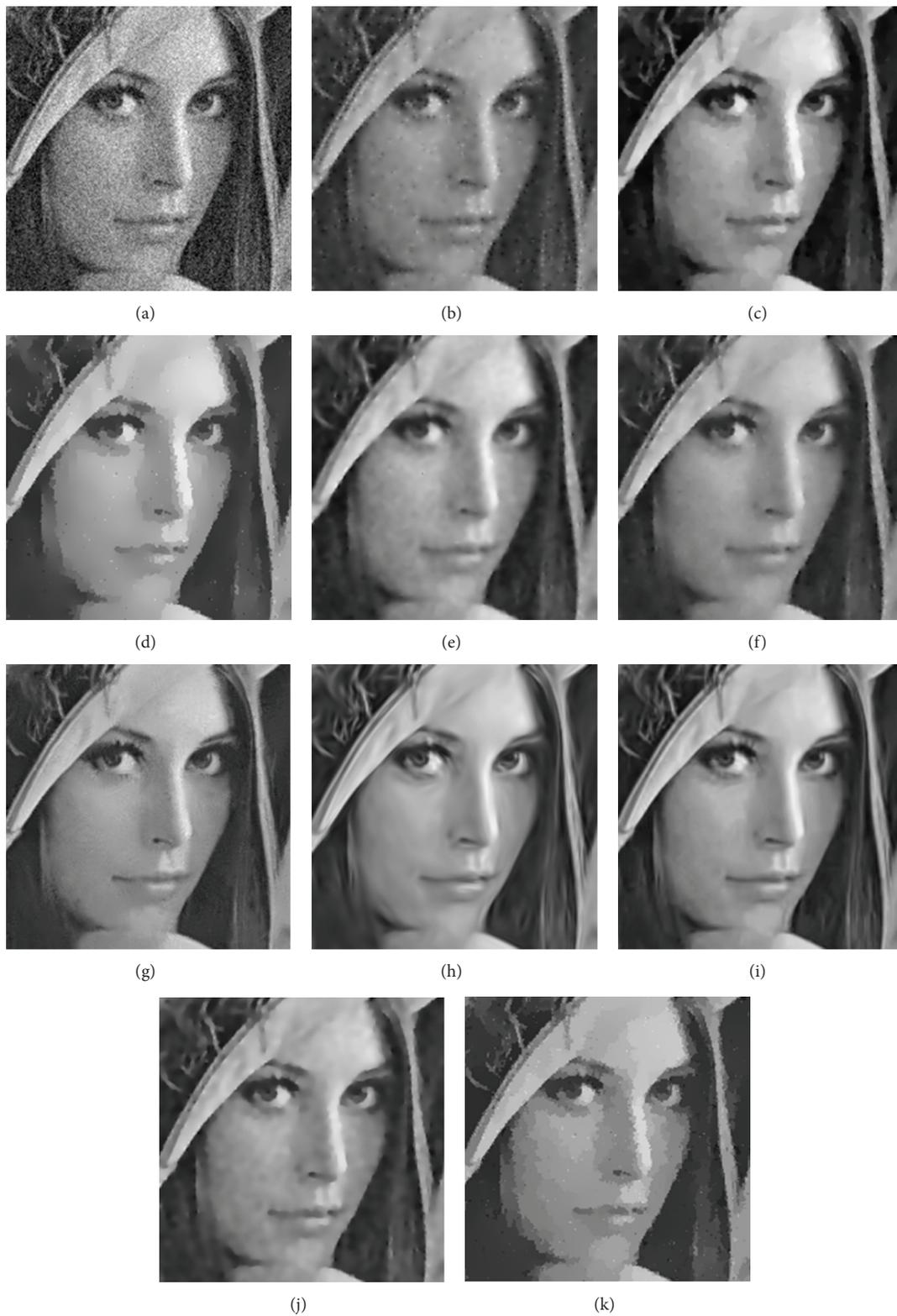


FIGURE 4: Demonstration and comparison of the Lena image. (a) Noisy image, filtered images by (b) the You-Kaveh fourth-order model, (c) the TV model, (d) the PM model, (e) the ATV model, (f) the STV model, (g) the NLM model, (h) the LARK model, (i) the BM3D model, (j) Hajiaboli's model, and (K) the DED model.

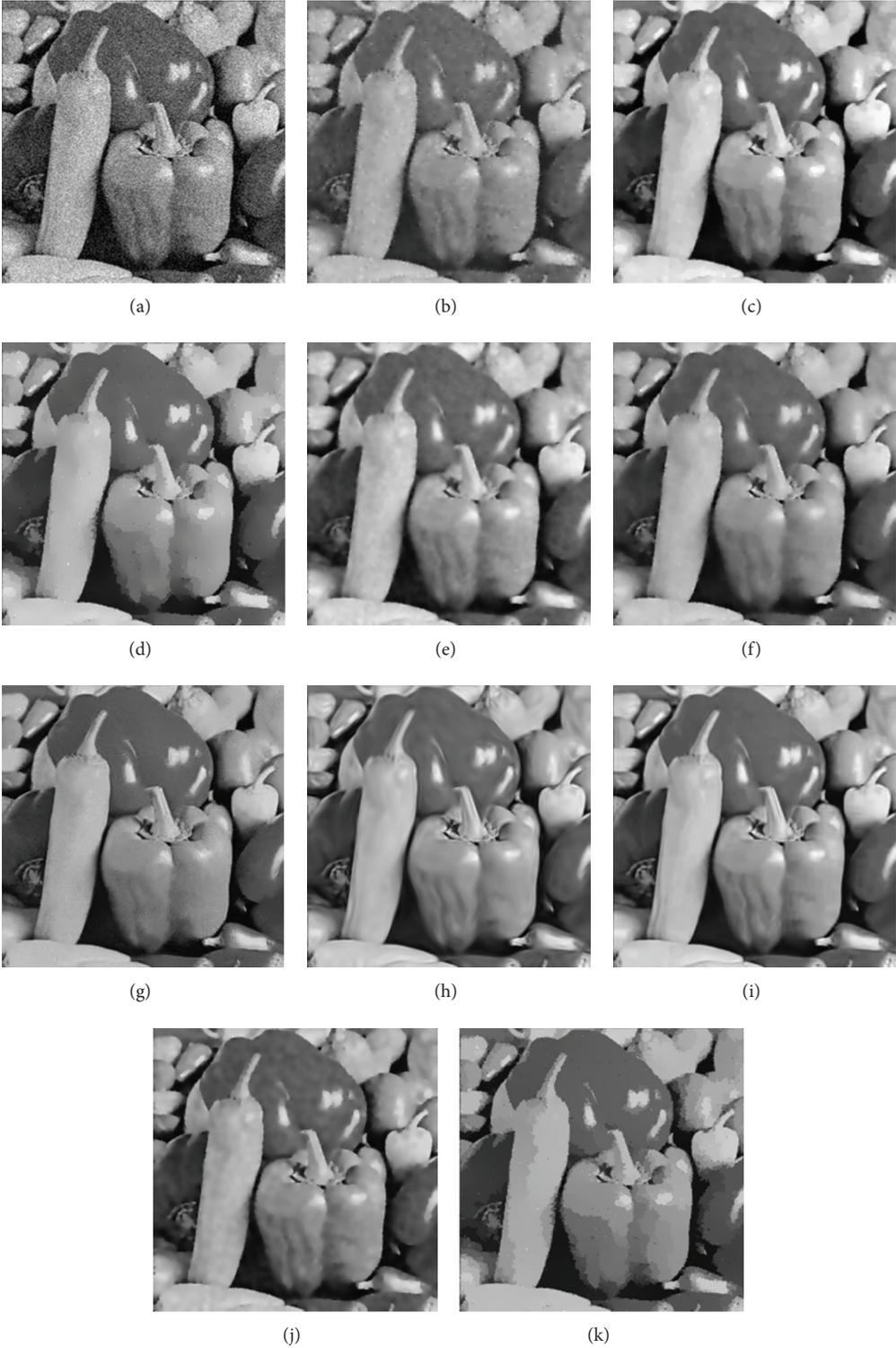


FIGURE 5: Demonstration and comparison of the pepper image. (a) Noisy images, filtered images by (b) the YK model, (c) the TV model, (d) the PM model, (e) the ATV model, (f) the STV model, (g) the NLM model, (h) the LARK model, (i) the BM3D model, (j) Hajiaboli's model, and (K) the DED model.

these models from the running time, PSNR and MSSIM indices, and visual inspection. Overall, the proposed STV model yields promising results and we believe that the local energy derived from the steerable filter acts as an effective indicator to discriminate edges and ramps and it can also be used in other PDE based approaches for image filtering.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to express their gratitude to the editor and anonymous reviewers for their constructive comments that lead to this paper's improvements in quality and representation. This work was supported in part by the Chinese Society of Neuro-oncology, Neuro-oncology Research Project CSNO-2013-MSD003, and the Program from the Tianjin Commission of Technology of China under Grant 11JCZDJC15600.

References

- [1] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.
- [2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the IEEE 6th International Conference on Computer Vision*, pp. 839–846, January 1998.
- [3] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 3, pp. 60–65, June 2005.
- [4] P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1635–1649, 2012.
- [5] K. Dabov, A. Foi, V. Katkovich, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [6] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, vol. 147, Springer, New York, NY, USA, 2nd edition, 2006.
- [7] P. Milanfar, "A tour of modern image filtering: new insights and methods, both practical and theoretical," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, 2013.
- [8] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [9] H. Tian, H. Cai, J. H. Lai, and X. Xu, "Effective image noise removal based on difference eigenvalue," in *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP '11)*, pp. 3357–3360, Brussels, Belgium, September 2011.
- [10] H. Wang, Y. Wang, and W. Ren, "Image denoising using anisotropic second and fourth order diffusions based on gradient vector convolution," *Computer Science and Information Systems*, vol. 9, no. 4, pp. 1493–1511, 2012.
- [11] T. Liu and Z. Xiang, "Image restoration combining the second-order and fourth-order PDEs," *Mathematical Problems in Engineering*, vol. 2013, Article ID 743891, 7 pages, 2013.
- [12] Y. Wang, W. Ren, and H. Wang, "Anisotropic second and fourth order diffusion models based on convolutional virtual electric field for image denoising," *Computers & Mathematics with Applications*, vol. 66, no. 10, pp. 1729–1742, 2013.
- [13] Y. Wang, J. C. Guo, W. F. Chen, and W. Zhang, "Image denoising using modified Perona-Malik model based on directional Laplacian," *Signal Processing*, vol. 93, no. 9, pp. 2548–2558, 2013.
- [14] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, 1992.
- [15] P. Blomgren and T. F. Chan, "Color TV: total variation methods for restoration of vector-valued images," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 304–309, 1998.
- [16] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.
- [17] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005.
- [18] M. Nikolova, "Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers," *SIAM Journal on Numerical Analysis*, vol. 40, no. 3, pp. 965–994, 2002.
- [19] H. Zhang and Y. Wang, "Edge adaptive directional total variation," *IET the Journal of Engineering*, vol. 1, no. 1, 2 pages, 2013.
- [20] T. F. Chan and S. Esedoglu, "Aspects of total variation regularized L^1 function approximation," *SIAM Journal on Applied Mathematics*, vol. 65, no. 5, pp. 1817–1837, 2005.
- [21] Y. Wang, W. Chen, S. Zhou, T. Yu, and Y. Zhang, "MTV: modified total variation model for image noise removal," *Electronics Letters*, vol. 47, no. 10, pp. 592–594, 2011.
- [22] M. Lysaker, A. Lundervold, and X.-C. Tai, "Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1579–1590, 2003.
- [23] Y.-L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1723–1730, 2000.
- [24] M. R. Hajiaboli, "An anisotropic fourth-order diffusion filter for image noise removal," *International Journal of Computer Vision*, vol. 92, no. 2, pp. 177–191, 2011.
- [25] P. Blomgren, T. F. Chan, P. Mulet, and C. K. Wong, "Total variation image restoration: numerical methods and extensions," in *Proceedings of the International Conference on Image Processing*, vol. 3, pp. 384–387, October 1997.
- [26] J. Savage and K. Chen, "On multigrids for solving a class of improved total variation based PDE models," in *Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems*, CMA, Oslo, Norway, August 2005.
- [27] P. Blomgren, *Total variation methods for restoration of vector valued images [Ph.D. thesis]*, UCLA, 1998.
- [28] T. Karkkainen and K. Majava, "Semi-adaptive optimization methodology for image denoising," *IEE Proceedings*, vol. 152, no. 1, pp. 553–560, 2005.

- [29] Q. Chen, P. Montesinos, Q. S. Sun, P. A. Heng, and D. S. Xia, "Adaptive total variation denoising based on difference curvature," *Image and Vision Computing*, vol. 28, no. 3, pp. 298–306, 2010.
- [30] S.-H. Lee and J. K. Seo, "Noise removal with Gauss curvature-driven diffusion," *IEEE Transactions on Image Processing*, vol. 14, no. 7, pp. 904–909, 2005.
- [31] A. Buades, B. Coll, and J.-M. Morel, "The staircasing effect in neighborhood filters and its solution," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1499–1505, 2006.
- [32] D. Bertaccini, R. Chan, S. Morigi, and F. Sgallari, "An adaptive norm algorithm for image restoration," in *Scale Space and Variational Methods in Computer Vision*, vol. 6667 of *Lecture Notes in Computer Science*, pp. 194–205, Springer, Berlin, Germany, 2012.
- [33] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.
- [34] W. T. Freeman, *Steerable filter and local analysis of image structure [Ph.D. thesis]*, MIT, 1992.
- [35] M. Kass and A. Witkin, "Analyzing oriented patterns," *Computer Vision, Graphics and Image Processing*, vol. 37, no. 3, pp. 362–385, 1987.
- [36] H. Knutsson and G. H. Granlund, "Texture analysis using two-dimensional quadrature filters," in *Proceedings of the IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management (CAPAIDM '83)*, pp. 206–213, 1983.
- [37] S. W. Zucker, "Early orientation selection: tangent fields and the dimensionality of their support," *Computer Vision, Graphics, and Image Processing*, vol. 32, no. 1, pp. 74–103, 1985.
- [38] Y.-L. You, W. Xu, A. Tannenbaum, and M. Kaveh, "Behavioral analysis of anisotropic diffusion in image processing," *IEEE Transactions on Image Processing*, vol. 5, no. 11, pp. 1539–1553, 1996.
- [39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

Research Article

2D Barcode Image Decoding

Jeng-An Lin and Chiou-Shann Fuh

Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan

Correspondence should be addressed to Chiou-Shann Fuh; fuh@csie.ntu.edu.tw

Received 17 June 2013; Revised 25 October 2013; Accepted 16 November 2013

Academic Editor: Chung-Hao Chen

Copyright © 2013 J.-A. Lin and C.-S. Fuh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Barcode technology is one of the most important parts of Automatic Identification and Data Capture (AIDC). Quick Response code (QR code) is one of the most popular types of two-dimensional barcodes. How to decode various QR code images efficiently and accurately is a challenge. In this paper, we revise the traditional decoding procedure by proposing a serial of carefully designed preprocessing methods. The decoding procedure consists of image binarization, QR code extraction, perspective transformation and resampling, and error correction. By these steps, we can recognize different types of QR code images. The experiment results show that our method has better accuracy than Google open-source 1D/2D barcode image processing library Zxing-2.1. Moreover, we evaluate the execution time for different-size images. Our method can decode these images in real time.

1. Introduction

Barcode technology is one of the most important parts of Automatic Identification and Data Capture (AIDC); we can obtain decoded data through analysis of a barcode. According to the encoding type of barcodes, we divide barcode into two categories: one-dimensional barcode and two-dimensional barcode. One-dimensional barcode typically consists of varying the widths and spacings of parallel lines. Moreover, two-dimensional barcode is a graphical image that stores information both horizontally and vertically. Two-dimensional barcode compared with the one-dimensional barcode has the following advantages: (1) high data capacity; (2) no additional storages; (3) error correction ability.

Quick Response code (QR code) is one of the most popular types of two-dimensional barcodes developed in Japan by Denso Corporation in 1994. Although QR code was first designed for the automotive industry, QR codes are now used over much wider range of applications, including commercial tracking, transport ticketing, website Uniform Resource Locator (URL), and identity verification. QR code has several advantages. First, QR code has strong error correcting capability, which can restore 30% data for the maximum error correction level. Second, QR code can be scanned from any direction because the proportion of

position detection patterns is not changed with the scanning direction. Third, QR code supports different encoding types and versions. We can choose an appropriate encoding type and version to reduce the size of QR code image.

Although there exist many commercial programs for QR code decoding, inventing more robust QR code decoding strategy still attracts many attentions from researchers. Many works are proposed recently. In [1, 2], Belussi and Hirata proposed a fast component-based two-stage approach for detecting QR codes in arbitrarily acquired images. Moreover Ohbuchi et al. [3] and Liu et al. [4] focused on enabling mobile phone to recognize QR code under poor conditions in real time. For decoding QR code image correctly and efficiently, we confront some challenges as follows. (1) When we capture QR code images using a phone camera in nonuniform background or uneven light, it might affect the decoded result. (2) The shooting angle might cause perspective distortion; to restore a distorted QR image is an important issue for decoding QR code images. (3) Decoding QR code images needs a sequence of sophisticated image processing. How to reduce the complexity of image processing and maintain the decoding accuracy rate is also a challenge.

In this paper, we focus on revising preprocessing methods in the traditional decoding procedure that is straightforward but still have much improving potential to overcome

the above challenges. The decoding steps consist of image binarization, QR code extraction, perspective transformation and resampling, and error correction. In image binarization, we use a modified local threshold method to resolve the nonuniform background and uneven light problems. Our QR code extraction algorithm uses the position detection patterns of QR code efficiently and accurately to locate the QR code position. After locating the QR code position, we restore a distorted image through perspective transformation and resample. Finally, we get the corrected data by Reed-Solomon error correction algorithm.

2. Related Work

QR code recognition technology has been studied in past years. We can simply divide the recognition into two steps, image preprocessing and QR code extraction.

In image preprocessing, some researchers focus on image denoising [5] or camera shaking [6]. In order to improve the performance of low resolution QR-code recognition, a previous work [7] uses the super-resolution technique that generates a high resolution image from multiple low-resolution images. Moreover, some researchers use different binarizations to improve the nonuniform background and uneven light problems [8, 9].

In QR code extraction, researchers propose several different methods to locate and extract the QR code of images. Some researchers use the feature of finder pattern to find rough QR code position [10–12]. After estimating the QR code's four corners using the rough QR code position, they effectively extract the QR code of images. In research [6, 13], they use edge detection to find possible rough barcode area. Then, morphological dilation and closing are used to generate more compact regions. Finally, the position of QR code of images can be detected.

Although these researches have their contributions, there are some shortcomings we can improve. In [5], researchers only focus on image denosing, but locating QR code position is an important part of QR decoding. They did not propose their method to process this problem. Previous works [6, 7, 10] need enormous calculation. Their methods are difficult to decode QR code image in real time. In research [9], their binarization method needs to know the version of QR code in advance. In this paper, we propose a decoding method to improve the deficiencies in previous work.

3. Background

3.1. Perspective Transformation. QR code images may be distorted due to camera perspective projection. Therefore, we use perspective transformation to restore the distorted QR code images. The general representation of a perspective transformation [14] is

$$\begin{bmatrix} x' & y' & w' \end{bmatrix} = \begin{bmatrix} u & v & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad (1)$$

where $x = x'/w'$ and $y = y'/w'$. In (1), each original coordinate (u, v) can be transformed to a new coordinate (x, y) by using the 3 by 3 matrix. Based on (1), x and y can be written as

$$\begin{aligned} x &= \frac{x'}{w'} = \frac{a_{11}u + a_{21}v + a_{31}}{a_{13}u + a_{23}v + a_{33}}, \\ y &= \frac{y'}{w'} = \frac{a_{12}u + a_{22}v + a_{32}}{a_{13}u + a_{23}v + a_{33}}. \end{aligned} \quad (2)$$

Without loss of generality, the 3 by 3 matrix can be normalized, so that $a_{33} = 1$. Equation (2) can be rewritten as

$$\begin{aligned} x &= a_{11}u + a_{21}v + a_{31} - a_{13}ux - a_{23}vx, \\ y &= a_{12}u + a_{22}v + a_{32} - a_{13}uy - a_{23}vy. \end{aligned} \quad (3)$$

Applying four pairs of correspondence points to (3), we can get a linear system as follows:

$$\begin{bmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 & -u_0x_0 & -v_0x_0 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -v_3x_3 \\ 0 & 0 & 0 & u_0 & v_0 & 1 & -u_0y_0 & -v_0y_0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3y_3 & -v_3y_3 \end{bmatrix} \times \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{13} \\ a_{23} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}. \quad (4)$$

The value of unknown coefficients will be determined by solving the linear system in (4).

3.2. Reed-Solomon Error Correction. Reed-Solomon codes are nonbinary cyclic error-correcting codes [15]. In QR code decoding procedures, we might obtain some error information because of distorted images or inappropriate binarization. Reed-Solomon codes help us to correct the error information. The general decoding algorithm [16] of RS codes consists of computing the syndromes, determining an error locator polynomial, and solving the error values.

For computing the syndromes, the received message is viewed as the coefficients of a polynomial $S(x)$:

$$S(x) = \sum_{i=0}^{n-1} r_i x^i, \quad (5)$$

where r_i is the received message at position i . Therefore, the syndromes $(S_0, S_1, \dots, S_{2t-1})$ are defined as

$$S_0 = S(\alpha^0), \quad S_1 = S(\alpha^1), \dots, S_{2t-1} = S(\alpha^{2t-1}), \quad (6)$$

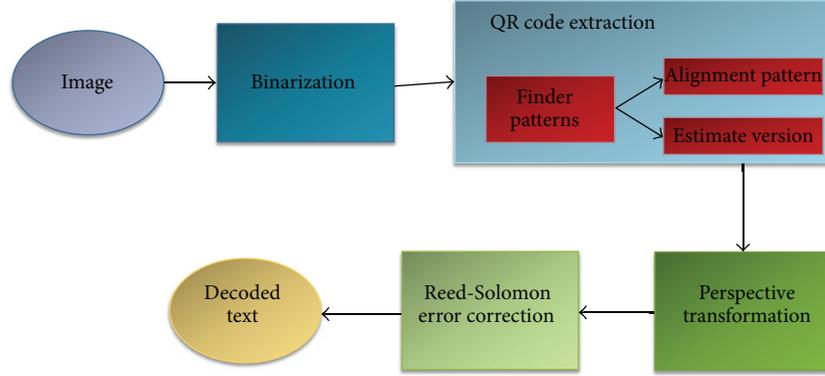


FIGURE 1: QR code decoding procedure.

where $2t$ is equal to the number of codewords available for error correction and α is the primitive element on the Galois field. Syndrome polynomial is defined as

$$S(x) = \sum_{i=0}^{2t-1} S_i x^i. \quad (7)$$

After computing syndrome polynomial, we use Euclidean algorithm to construct the error locator polynomial and error evaluator polynomial. Error locator polynomial can be used to find the error positions. Error evaluator polynomial is used to find the error values. The Euclidean algorithm is based on the key equation in

$$\Omega(x) = S(x) \Lambda(x) \pmod{x^{2t}}. \quad (8)$$

Given only $S(x)$ and t , we want to obtain the error locator polynomial $\Lambda(x)$ and the error evaluator polynomial $\Omega(x)$. Equation (8) means that

$$\theta(x)x^{2t} + S(x) \Lambda(x) = \Omega(x) \quad (9)$$

for some polynomial $\theta(x)$. Finally, we can use Euclidean algorithm to generate the error locator polynomial and error evaluator polynomial.

For solving the error values, Forney's algorithm is selected. The error values for a Reed-Solomon code are computed by

$$e_{i_k} = -\frac{\Omega(X_k^{-1})}{\Lambda'(X_k^{-1})}, \quad (10)$$

where e_{i_k} is the corrected value at position i_k ; X_k^{-1} is the root of $\Lambda(x)$ at position k ; and $\Lambda'(x)$ is the formal derivative of $\Lambda(x)$. By (10), we obtain the corrected data of QR code images.

4. The Proposed Method

We design a QR code recognition procedure for decoding QR code accurately and rapidly. First, we propose a new local

binarization method to overcome images with uneven light and nonuniform background problems. And then we use three finder patterns and an alignment pattern of QR code efficiently to locate the QR code position. After locating QR code position, we resample QR code image by perspective transformation. Finally, we get the corrected data by Reed-Solomon error correction algorithm. Figure 1 illustrates the procedure.

4.1. Binarization. Binarization is an important process for accurately recognizing black-and-white module in QR code images. Therefore, we propose a new binarization method to improve the recognition of images with nonuniform background and uneven light. Our method is a local threshold algorithm based on Sauvola's method [8]. We decompose each image into a series of blocks and then calculate a threshold T_1 for each block using

$$T_1(x, y) = \text{mean}(x, y) \times \left[1 + \left(\frac{\text{standard deviation}(x, y)}{R} \right) \right]. \quad (11)$$

The default R is 1250. For each block, we convolve with the kernel in (12) to calculate a threshold T_2 . By using the threshold T_2 , we split each block into bright area and dark area. Figures 2 and 3 shows that our method can achieve the same performance with less computing cost, compared with modified Sauvola's method [8]:

$$C = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad (12)$$

$$f(A, B, C) = (B \cdot x - A \cdot x)(C \cdot y - A \cdot y) - (B \cdot y - A \cdot y)(C \cdot x - A \cdot x). \quad (13)$$

4.2. Estimating Version. QR code extraction is responsible for searching QR code area in images. QR code extraction consists of searching finder patterns, estimating version, and



FIGURE 2: An uneven light image [17].



FIGURE 3: Binary image using our method.

searching alignment pattern. We use the proportion of finder patterns to locate the QR code position. After searching three finder patterns, we use the distance between these patterns to estimate version. Finally, we develop a connected-component method to search alignment pattern accurately.

4.2.1. Searching Finder Patterns. In searching finder patterns, we need to determine the upper left, lower left, and upper right finder patterns in the QR code. First, we scan the binary image horizontally to find all points matching the ratios 1:1:3:1:1. For each point matching the ratios, we check its vertical direction. Finally, we have collected all points in images matching the ratios 1:1:3:1:1 both horizontally and vertically.

After collecting points, we merge points in the same finder pattern. If the number of finder patterns is more than three, we filter out some inappropriate points. First, we calculate the angle between three points. If the maximum angle is greater than a threshold T_1 or the minimum angle is less than a threshold T_2 , the points will be filtered. And then we identify the direction of finder patterns. The maximum angle corresponding to the finder pattern is the upper left



FIGURE 4: Red line is the width of finder pattern, and blue line is the estimated width.

finder pattern. Finally, we use cross product in (13) to determine the lower left and upper right finder patterns, where A is the upper left finder pattern; B and C are the other finder patterns; and $A \cdot x$ and $A \cdot y$ represent the x and y coordinate of A . If $f(A, B, C) < 0$, then B is upper right finder pattern, and C is lower left finder pattern.

4.2.2. Estimating Version. After searching three finder patterns, we can estimate version using module size and the distance between two finder patterns. Module size can be calculated by

$$\text{Module_size} = \frac{\text{Width of finder pattern}}{7}. \quad (14)$$

We calculate version by

$$\begin{aligned} \text{Version} &= \frac{((\text{Euclidean_distance}(f_1, f_2) / \text{Module_size}) - 10)}{4}, \end{aligned} \quad (15)$$

where f_1 is upper left finder pattern and f_2 is upper right finder pattern. We can get rough version by (15), but (15) misjudges the version when QR code image is rotated. We overestimate the width of finder pattern in rotated image in Figure 4.

As a result, we underestimate the version of QR code image. Therefore, we revise (15) to be

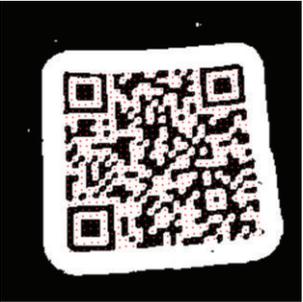
$$\begin{aligned} \text{Version} &= \frac{(\text{Euclidean_distance}(f_1, f_2) / \text{Module_size} * \cos \theta - 10)}{4}, \end{aligned} \quad (16)$$

where θ is the angle between blue line and red line in Figure 4. We estimate version more accurately by (16) and thus before perspective transformation.

TABLE 1: Eight coordinates used for perspective transformation ($D = 17 + 4^*$ version).

Function pattern of QR code	Original coordinate	Normalized coordinate
Upper left finder pattern	(x_0, y_0)	$(3.5, 3.5)$
Upper right finder pattern	(x_1, y_1)	$(D-3.5, 3.5)$
Lower left finder pattern	(x_2, y_2)	$(3.5, D-3.5)$
Lower right alignment pattern	(x_3, y_3)	$(D-6.5, D-6.5)$

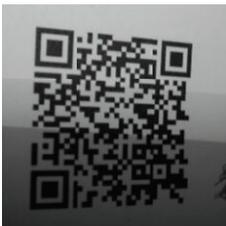
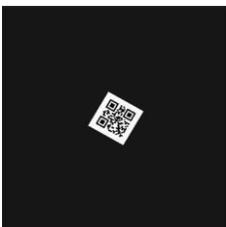
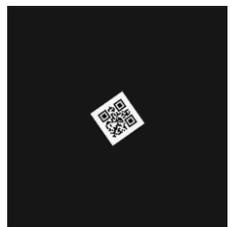
TABLE 2: Result of resampling.

Original image	Normalized image
	
	
	

4.3. *Searching Alignment Pattern.* Low resolution image or inappropriate binarization method will result in the deformation of alignment pattern. Therefore, we propose a new searching alignment method to overcome the problem. First, we roughly locate the alignment pattern using three finder patterns. Moreover, we extract subimage with a specified radius in Figure 5. We calculate the connected component in the subimage, both black pixels and white pixels. After calculating the connected component, we get two images in Figures 6(b) and 6(c).

For each connected component of black pixel C_i in Figure 6(b), we check its border pixels. If all border pixels are

TABLE 3: Test images.

Set 1		
Set 2		
Set 3		
Set 4		
Set 5		
Set 6		

adjacent to the same connected component of white pixel C'_j in Figure 6(c), then we check all border pixels in C'_j . Once all border pixels in C'_j are adjacent to the same black pixel connected component C_k in Figure 6(b), these components possibly contain alignment pattern. Finally, we calculate the centroid of C_i as the centroid of alignment pattern.

TABLE 4: Comparison between our method and Zxing-2.1.

	Our method	Zxing-2.1
Set 1	20/20	17/20
Set 2	32/34	30/34
Set 3	38/42	38/42
Set 4	36/48	36/48
Set 5	19/19	19/19
Set 6	15/15	15/15
Total	160/178	155/178

TABLE 5: Images we successfully recognize and Zxing-2.1 fails.



4.4. Perspective Transformation. After estimating the version and locating the centroid of three finder patterns and alignment pattern, we resample QR code image through perspective transformation. The eight coordinates can be presented as shown in Table 1.

To avoid the appearance of cracks and holes, we use inverse warping in perspective transformation. First, we calculate a transformation T from normalized coordinate to original coordinate. For each pixel in normalized image, we calculate its corresponding location in original image using the transformation T . Finally, we copy the pixel value of original image to the normalized image. Table 2 shows our resampling results.



FIGURE 5: We extract blue rectangle to find the alignment pattern (radius = 30 pixels).

5. Experiment Result

Experimental environment:

- (1) CPU: Intel core i5-3210M CPU 2.50 GHz,
- (2) memory: 4 GB,
- (3) OS: Windows 7,
- (4) programming language: Java.

In this section, we conduct several experiments to evaluate the performance. First, we compare the accuracy between our QR code decoding algorithm and 2D barcode image processing library (Zxing) [17]. The experiment result shows that our decoding algorithm has better accuracy than Zxing. And then we use different sizes of images to evaluate the decoding time. The experiment results show our method decoding images in real time. Moreover, we implement a QR code scanner on Android phone. We test the decoding ability of some defaced images. The experiment result shows that our QR code scanner successfully decodes these images.

5.1. Image Source. Our test images in Table 3 are from Google open-source 1D/2D barcode image processing library Zxing-2.1 [17]. The images are classified into six data sets, a total of 178 images. The first data set contains low resolution QR code images. The second data set contains trademark QR code image. The skew QR code images are included in the third data set. The fourth data set contains distorted QR code images. Uneven light QR code images are included in the fifth data sets. The sixth data set contains rotated QR code images.

5.2. Accuracy Analysis. We compare the accuracy between our decoding algorithm and Google open-source 1D/2D barcode image processing library Zxing-2.1 [17]. Table 4 shows the results.

In 178 test images, there are 8 images in Table 5 that we successfully recognize and Zxing-2.1 fails. Moreover, there are 3 images in Table 6 that Zxing-2.1 successfully recognizes and our method fails.

The accuracy of our method is about 90% and that of Zxing-2.1 is only 87%. Our method has better recognition rate

TABLE 6: Images Zxing-2.1 successfully recognizes and our method fails.

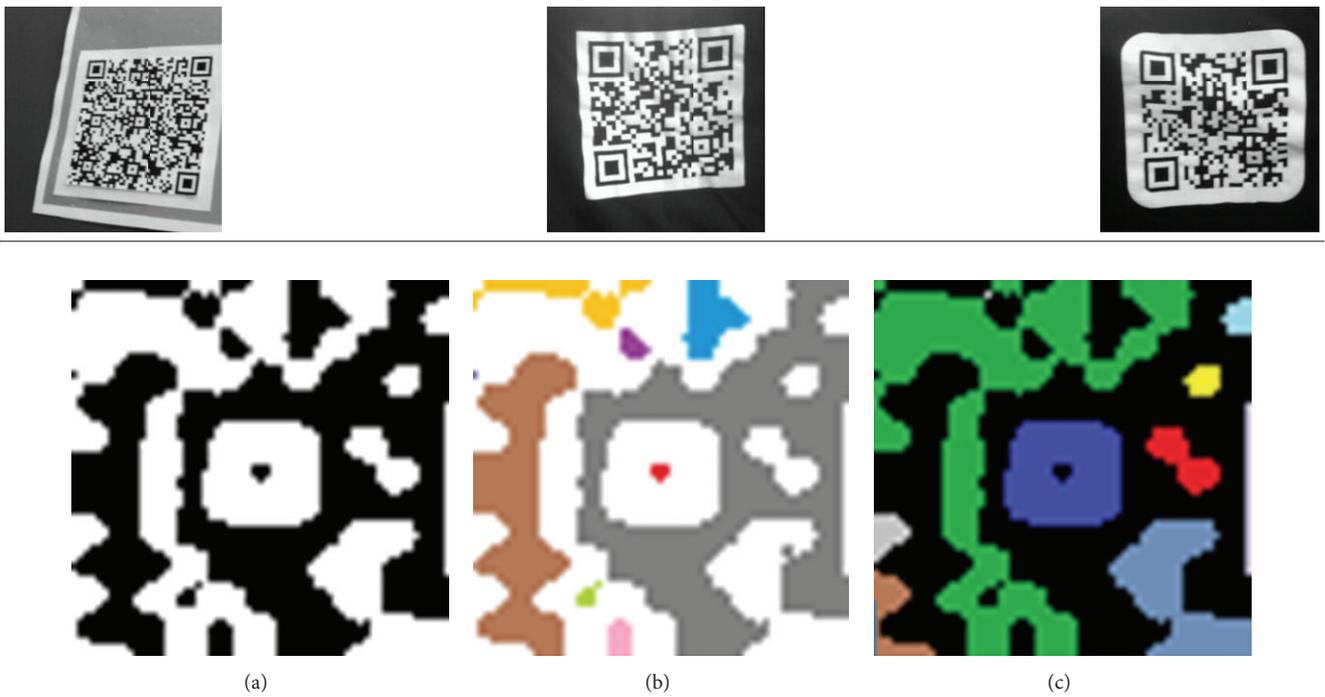


FIGURE 6: (a) Extracted image in Figure 5; (b) found connected components of black pixels; (c) found connected components of white pixels.

TABLE 7: The execution time between our method and Zxing for decoding different-size images.

	Our method	Zxing
320 * 240	214 ms	139 ms
640 * 480	312 ms	190 ms
800 * 600	375 ms	237 ms
1024 * 768	448 ms	264 ms

in Sets 1 and 2. Zxing-2.1 cannot accurately locate the alignment pattern because of the low resolution or inappropriate binarization. Our method improves the problem, so that we have better result.

5.3. Efficiency Analysis. In this section, we show the execution time between our method and Zxing for decoding different sizes of QR code images. We use four different sizes of QR code images to calculate the execution time. Each size contains five images. Table 7 shows the results.

Our method needs more decoding time than Zxing, but our method still can decode QR code images in real time. We only need 448 milliseconds for decoding resolution 1024 * 768 pixel images.

5.4. Comparison of Different Barcode Applications. We implement a QR code scanner on Android phone. Moreover, we collect four popular barcode scanners from Google Play and iOS market. We compare the decoding ability of some

defaced images. Table 8 shows the result. Our application can successfully decode all test images. Our application has the best result of these applications in these images. The reason for our better results is that two steps of the binarization stage in our method will get a more appropriate threshold to split each block into bright area and dark area. Moreover compared with current applications, our new searching alignment method could handle a large deformation of alignment pattern as the result of inappropriate binarization and low resolution which can be observed in sample images.

6. Conclusion

In this paper, our carefully designed preprocessing methods improve the accuracy of the traditional decoding procedure. Although refined steps improve accuracy at the expense of increased decoding time [21], the QR code still can meet the real-time requirement. The decoding steps consist of image binarization, QR code extraction, perspective transformation, and error correction. We efficiently resolve the non-uniform background and uneven light problems by using a modified local threshold algorithm. In QR code extraction, we use the position detection patterns of QR code efficiently and accurately to locate the QR code position. After locating the QR code position, we restore a distorted image through perspective transformation and resampling. Finally, we get the corrected data by Reed-Solomon error correction algorithm.

In the experiment, we compare the accuracy between our method and Zxing. The experiment result shows our

TABLE 8: Comparison of different barcode scanner applications.

	Our	Zxing [17]	Quickmark [18]	Bakodo [19]	BarCodeScan [20]
	O	O	X	X	X
	O	X	X	X	X
	O	X	O	O	O
	O	X	O	O	O
	O	O	O	O	X
	O	O	O	O	X

TABLE 8: Continued.

	Our	Zxing [17]	Quickmark [18]	Bakodo [19]	BarCodeScan [20]
	O	X	O	O	O
	O	X	O	O	O
	O	X	X	X	X
	O	X	X	X	X
Total	10/10	3/10	6/10	6/10	4/10

method to have better accuracy than Zxing. Moreover, we use different-size images to calculate the execution time. The experiment shows that our method can efficiently decode images. All images can be decoded in less than one second.

References

- [1] L. F. F. Belussi and N. S. T. Hirata, "Fast component-based QR code detection in arbitrarily acquired images," *Journal of Mathematical Imaging and Vision*, vol. 45, no. 3, pp. 277–292, 2013.
- [2] L. F. F. Belussi and N. S. T. Hirata, "Fast QR code detection in arbitrarily acquired images," in *Proceedings of the Conference on Graphics, Patterns and Images (SIBGRAPI '11)*, pp. 281–288, Maceió, Brazil, August 2011.
- [3] E. Ohbuchi, H. Hanaizumi, and L. A. Hock, "Barcode readers using the camera device in mobile phones," in *Proceedings of the International Conference on Cyberworlds (CW '04)*, pp. 260–265, Tokyo, Japan, November 2004.
- [4] Y. Liu, J. Yang, and M. Liu, "Recognition of QR Code with mobile phones," in *Proceedings of the Chinese Control and Decision Conference (CCDC '08)*, pp. 203–206, Yantai, China, July 2008.
- [5] C. C. Lin and M. S. Chen, *A general scheme for QR-code image denoising on the camera phone [M.S. thesis]*, Department of Electrical Engineering, National Taiwan University, 2009.
- [6] C.-H. Chu, D.-N. Yang, and M.-S. Chen, "Image stabilization for 2D barcode in handheld devices," in *Proceedings of the 15th ACM International Conference on Multimedia (MM '07)*, pp. 697–706, September 2007.
- [7] Y. Kato, D. Deguchi, T. Takahashi, I. Ide, and H. Murase, "Low resolution QR-code recognition by applying super-resolution using the property of QR-codes," in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR '11)*, pp. 992–996, September 2011.
- [8] J. J. Zhou, Y. F. Liu, and P. Li, "Research on binarization of QR code image," in *Proceedings of the International Conference on Multimedia Technology (ICMT '10)*, pp. 1–4, Ningbo, China, October 2010.
- [9] Z. Xiong, H. Cuiqun, L. Guodong, and L. Zhijun, "A binarization method of quick response code image," in *Proceedings of the 2nd International Conference on Signal Processing Systems (ICSPS '10)*, vol. 3, pp. 317–320, Dalian, China, July 2010.
- [10] Y. H. Chang, C. H. Chu, and M. S. Chen, *A general scheme for extracting QR code from a non-uniform background in camera*

- phones and applications [M.S. thesis]*, Department of Electrical Engineering, National Taiwan University, 2007.
- [11] C. Y. Lai and M. S. Chen, *Extracting QR Code from a non-uniform background image in embedded mobile phones [M.S. thesis]*, Department of Electrical Engineering, National Taiwan University, 2007.
- [12] Y. L. Pan and M. S. Chen, *2D barcodes from images with spatial distortion [M.S. thesis]*, Department of Electrical Engineering, National Taiwan University, 2008.
- [13] D. Munoz-Mejias, I. Gonzalez-Diaz, and F. Diaz-De-Maria, "A low-complexity pre-processing system for restoring low-quality QR code images," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1320–1328, 2011.
- [14] P. S. Heckbert, *Fundamentals of texture mapping and image warping [M.S. thesis]*, Department of Electrical Engineering, University of California, Berkeley, Calif, USA, 1989.
- [15] Wikipedia, "Reed-Solomon Error Correction," 2013, http://en.wikipedia.org/wiki/Reed-Solomon_error_correction.
- [16] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*, Wiley, New York, NY, USA, 2005.
- [17] S. Owen, "Zxing ("Zebra Crossing")," 2013, <https://code.google.com/p/zxing/>.
- [18] SimpleAct, "QuickMark," 2013, <http://www.quickmark.com.tw/>.
- [19] A. Dougl, "Bakodo," 2013, <http://bako.do/>.
- [20] P. Giudicelli, "Barcode-Scanner," 2013, <https://itunes.apple.com/tw/app/barcode-scanner/id478602171?mt=8>.
- [21] T. Falas and H. Kashani, "Two-dimensional bar-code decoding with camera-equipped mobile phones," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '07)*, pp. 597–600, New York, NY, USA, March 2007.